FAKULTI SAINS KOMPUTER DAN TEKNOLOGI MAKLUMAT
UNIVERSITI MALAYA
KUALA LUMPUR

XML FOR BROWSER-BASED ELECTRONIC MEDICAL RECORD

HAFIZAM BIN MAT HUSSIN
WET 010031

WXES 3182: Projek Ilmiah Tahap Akhir II
Sesi 2004/2005

PENYELIA: PUAN SURAYA HAMID

MODERATOR: DR. TEH YING WAH

ABSTRACT

XML for Browser-based Electronic Medical Records is a system that using XML to represent, store and display information that is extracted from existing hospital systems. It is an XML database designed to provide a clinician-centered view of the patient's history, with a web browser as the user interface.

Patient records are created and maintained as XML documents within an XML database. Data feeds from the HIS make use of existing HIS interfaces with some customization to meet the data requirements of the XML patient record database. HIS extracts are triggered by events occurring within the system. As updates take place, the XML database can provide full auditing of transactions that have taken place.

The major modules in XML for Browser-based Electronic Medical Records are the administrator module and user module. This report focuses on the administrator module. The administrator module provides support for various administrative tasks, which are controlling user access, memo/message board management, search services and e-mail services.

The proposed development tools for this project will take advantage of one of the latest technologies, which is XML, Microsoft Access 2003 as the DBMS, Macromedia Dreamweaver MX as a development workspace, Windows NT server as the development platform, plus several other tools.

This system sees the opportunity in avoiding and rectifying the weaknesses found in existing electronic medical records systems or websites and hopes to provide a better solution to the medical field.

ACKNOWLEDGEMENT

Firstly, I would like to express my gratitude towards Puan Suraya binti Hamid for her supervision over the completion of this final year project. Her constructive and creative ideas plus the encouragement certainly help me a lot over this period of time.

Secondly, I would like to express my appreciation towards my project moderator, Dr. Teh Ying Wah, for pointing out the areas of this project and additional ideas, which could do with some further enhancement. Dr. Teh also brought my attention to specific matters that require more consideration and deeper understanding, especially during the viva session.

Thirdly, I would also like to take this opportunity to thank my partner of this project, Maizirah binti Ibrahim who contributed all out, as well as help me to build the final project until this ended point and hopefully we also can integrate together in other time.

I would also like to thank all my course mates, such as Raja Firdaus, Asroll and Nazir who contribute invaluable ideas for my project as well as help to look for solutions to my problems regarding this project, particularly Rasulizam Hasan, Mr Ramesh, Sazali, Wira, Zarir and all my friends for their significant work rate and contribution.

Last but not least, thanks to everyone who spared their valuable time for giving ideas regarding the requirements study of this project and also provided vital information related to XML and Electronic Medical Records.

Table of Contents **Page**

CHAPTER 8 : SYSTEM EVALUATION

List Of Figures:

List Of Tables

CHAPTER 1 INTRODUCTION

1.0 Project Definition

Although Malaysia is marching towards the vision 2020 and the MSC status, most of the hospitals or clinics in Malaysia still using manual system to manage patient's records. Manual system is less efficient and error prone compare to computer-based system. The transformation of manually to computer-based system will certainly ease the job of management as well as reducing errors, potential risks, waiting time as well as appropriate dispensation to computer-based system will certainly ease the job of management as well as reduced errors, potential risks, waiting time as well as appropriate dispensary.

Electronic Medical records is a system that is used to simplify day to day task in running a clinic group in a modern hospital at the same time provide management with the necessary information to make strategic decisions. It serves essentially as a medium for communication between the diverse collaborating functional subsystems in a group of clinic in a hospital.

Electronic medical record summaries are now being deployed, using XML to represent, store and display information that is extracted from existing hospital systems. Patient records are created and maintained as XML documents within an XML database. Data feeds from the Hospital's Information System (HIS) make use of the existing HIS interfaces with some customization to meet the data requirements of the XML patient

record database. HIS extracts are triggered by events occurring within the system. As updates take place, the XML database can provide full auditing of transactions that have taken place. Access to the browser application is controlled by a security layer that can be integrated with the HIS system. The Web server provides the middle layer between the user interacting with the browser and the XML database.

This system suits well six types of users; doctors, pharmacists, receptionist, account clerk and management staffs. Only authorized users are allowed to create, edit, delete, update or maintain the data under their respective scope. Validation of the user's login is indeed needs in order to access to the system. Functions provided for this system included managing patient records for a clinic group in a hospital, billings, and also report generation hence, it is hoped that this system will carry lot of benefits to all of the users besides reduce paperwork, working and storing space, the cost and redundant data entry and also increase productivity.

1.1 Project Objectives

Before any further planning or development is carried out, it is vital to draw out the objectives of the system, in order to provide a clearer picture of the requirements and also the needs of implementing the system. A number of objectives have been outlined for this system, which includes:

- **To keep the medical records in more secured way and eliminate the duplicated as well as redundant data.**

  All patient records are stored in a database. Only the authorized personnel are permitted to access the data. Medical records of patient can only be accessed by authorized personnel like doctors. So, there's no need to worry whether the medical report will get noticed by other people. Besides, user is alerted if patient ID or IC is already present when creating a new patient record. This help to avoid case medical records misplaced or lost. It will also eliminate the duplicated data in the database and also the common error with the manual system.

- **To archive the paperless administration.**

  Much paper work will be eliminated when all medical records are input to the computer and stored in the database.

- **To provide doctors safer and easier treatment or prescription writing.**

  The doctor can write patient's prescription into computer for future reference. No more paper or stationary used to note down the prescription. Doctors can be easily referred to patient's previous treatment and profiles. Pharmacists can refer to patient's prescription by clicking the mouse and keyboard to give suitable medicine to patient. It is safer than manual system because system eliminates common error with the manual system like prescription lost when passing from doctor to pharmacist.

- **To increase the efficiency of handling or retrieving medical records or data.**

  For the first time visit there is no record for the patient. The staff will do the registration for him or her. The staff will key in the patient's information and save it into database. The data of the patient may change from time to time; the staff can update the patient's profiles and save it again in the database. Furthermore, the patient's records can be retrieved easily and quickly. Before the consultation starts, doctors can easily achieve the patient's records through the database. This can avoid the doctors thumbing through several pieces of paper before he finds what he needs. So, the system will reduce the expenses and waiting time.

- **To create an appointment module.**

  Patients will be able to schedule their own appointment with the doctors and they will be informed if the doctor changes the appointment.

- **To produce detailed billing statement for patient's account.**

  This system will provide a more systematic way of handling the billing and payment system. A detailed statement will be produced and patient will be aware of what they are actually being accounted for during their stay or service at the hospital.

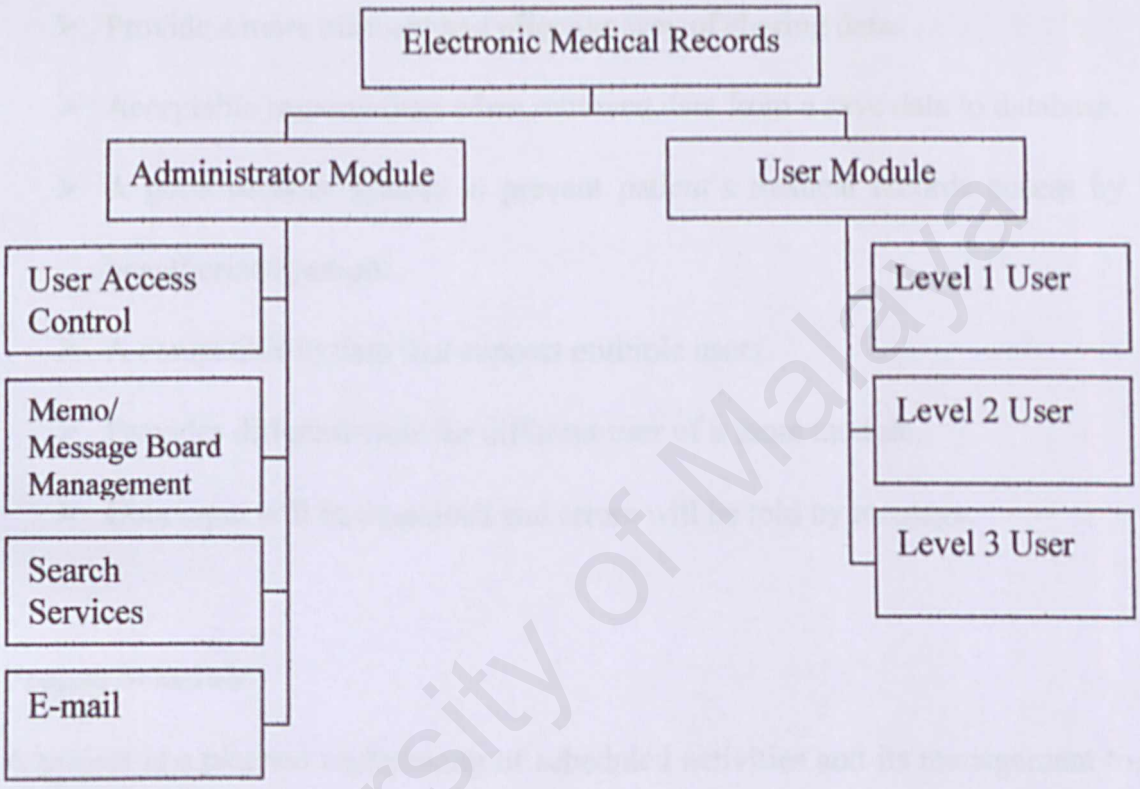
- **To produce standard medical reports.**

  The certificate produced in standard format and it is detailed, neat, and easy to read. It is available at any time required.

1.2 Scope and Limitations.

1.2.1 Scope of the project

The system is divided into two separate modules as depicted below:



Remark:
Level 1 User – Patient
Level 2 User – Receptionist, Account Clerk, Nurse, Pharmacist
Level 3 User – Doctor

Figure 1.0: The Project Scope

The scope that will cover by the system includes:

- Login to verify only the authorized person can access data under their respective scope.
- Develop the computerized system for patient registration.

- ❖ Develop a database system to store all related data.
- ❖ Develop a function that allow patient to make appointment.
- ❖ Generate billing statement for patient.
- ❖ Generate medical reports for reference and analysis purpose.

1.2.2 Limitations of the Project

The Electronic Medical Records system has some limitations as below:

- ➢ The appointment system is limited to patients who need further treatments from the hospital.
- ➢ Patient still using telephone or directly make an appointment rather than using the system itself or email.

1.2.3 Target User

Electronic medical records have its own scope of target users. They are those working in the hospitals and clinics. From the registration counter, the data entry clerks, nurses, doctors and physicians, pharmacist, account clerks, until the top-level personnel which include the record and also management personnel. Those must have permission to login. They are called authorized user and they will have their own username and password that will be requested before they enter the system. Besides them, the patients themselves will be able to access certain functions in the system such as their medical report, billing statement and can even make appointment with the physicians.

1.3 Expected Outcomes

Once an electronic Medical record is developed, it is expected to achieve the following features:

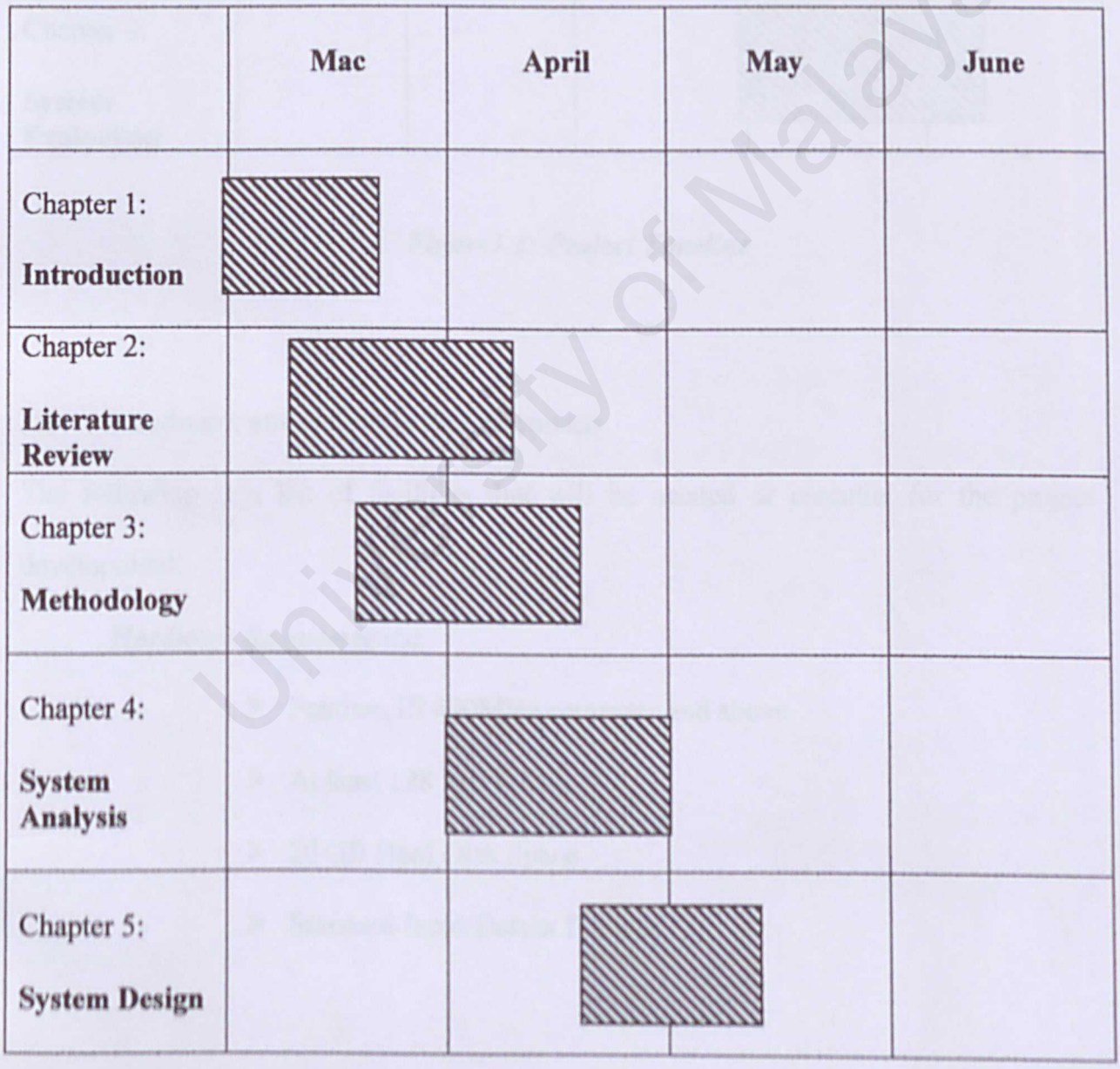
- User friendly and easy-to-learn system.
- Provide a more efficient and effective way of sharing data.
- Acceptable respond time when retrieved data from a save data to database.
- A good security system to prevent patient's medical records access by unauthorized person.
- A compatible system that support multiple users.
- Provides different view for different user of a same module.
- Data input will be examined and errors will be told by message.

1.4 Project Schedule

A project is a planned undertaking of scheduled activities and its management to reach a goal. Since a project may involve extensive effort, it must be property managed. Since electronic medical records is a final year project, which needs to be completed within a period of time. Planning is done to:

- Define the goals
- Define and allocate resources
- Establish timetable, schedule work
- Track and monitoring project
- Report and document the project

This project is divided into two phases, which will be referred to as Semester I and Semester II. During semester I, research on introduction, literature review and system analysis & design are carried out. In semester II, system coding, testing, evaluation and training will be done. Documentation is done from the beginning of the project until the end of the project.



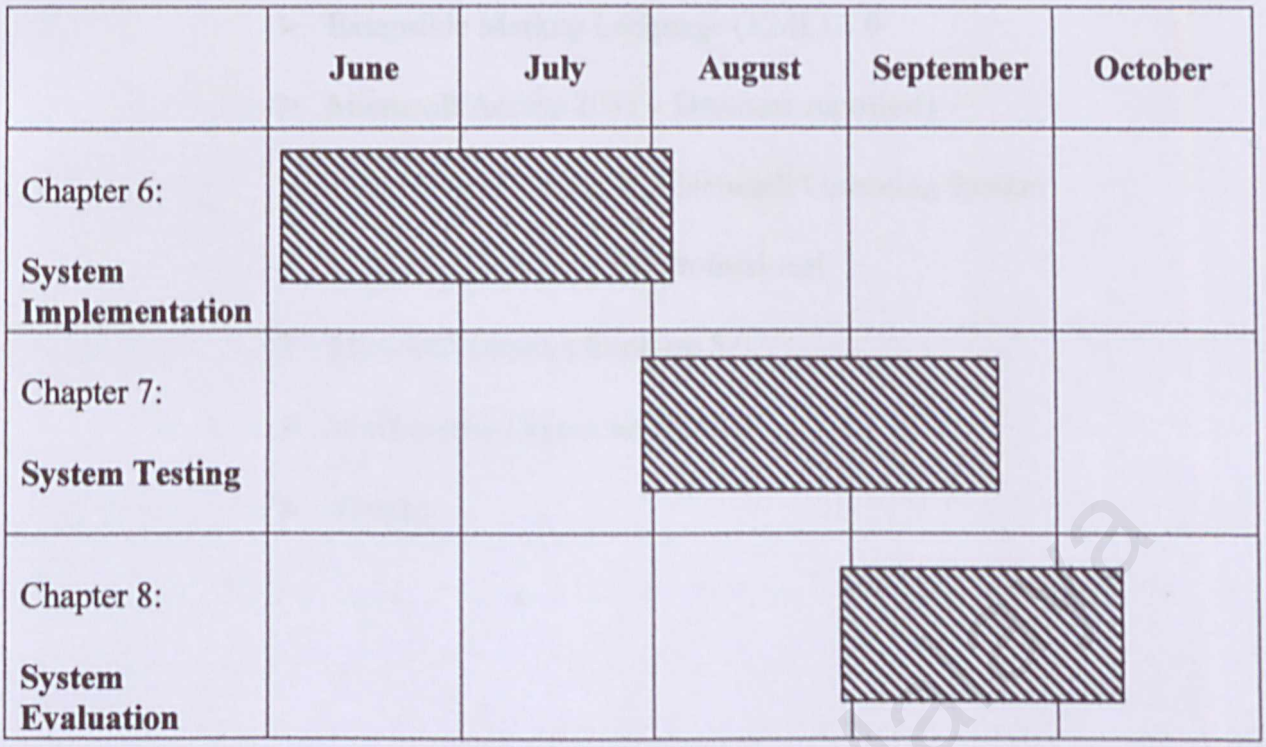


Figure1.1: Project Timeline

1.5 Hardware and Software Requirements

The following is a list of facilities that will be needed or essential for the project development:

Hardware Requirements:

- Pentium III 400MHz computer and above
- At least 128 MB RAM
- 20 GB Hard Disk Space
- Standard Input/Output Devices

Software Requirements:

- Extensible Markup Language (XML) 1.0
- Microsoft Access 2003 – Database repository
- Windows NT Server 4.0- Network Operating System
- Microsoft Windows XP Professional
- Microsoft Internet Explorer 5.0
- Macromedia Dream weaver MX
- Notepad

1.6 Overview of the chapter

Chapter 1: Introduction

This chapter consists of the project definition of electronic medical records, project objectives, scope and limitation project, expected outcomes, project schedule and hardware and software requirements.

Chapter 2: Literature Review

The literature review gives brief explanation on topics researched and studied that are relevant to this project. Anyway, the topics are definition of electronic medical records, comparison between manual and computer-based system, development tools consideration, and etc.

Chapter 3: Methodology

This chapter emphasizes on methodology been employed and information gathering techniques.

Chapter 4: System Analysis

This chapter consists of analysis of the functional requirement and non-functional requirement of the project. It also gives the explanation about the development software and platform chosen to develop this system. System requirement also listed in this chapter.

Chapter 5: System design

This system gives some detail about the design of the proposed system. In this chapter, there will be the functionality design, graphical user interface design and database design of the proposed system. This chapter explains the conceptual and technical design of the system. It covers the structure chart, data flow diagram, database design and user interface design.

1.7 Summary of chapter 1:

This chapter consists of the overview of the proposed project including the project definition and project objectives in general. Scope and limitations of the project also discussed in this chapter.

Other than that, the project also present an XML approach for represent, store and display information. Also include in this chapter are the target user, expected outcomes, project schedule and also hardware and software requirements.

CHAPTER 2 LITERATURE REVIEW

2.0 Introduction

This chapter is one of the most informative chapters. A lot of reading, reference, research and review have been made to gain information and basic knowledge before the system development process. It consists of definitions of the project's title and a brief description of some technology applications chosen to be considered in system's development process. We will also depict some examples of existing systems. Besides that, we made comparisons between the manual and computer-based system to show how beneficial this proposed system is.

2.1 Definitions

2.1.1 Definition of "Electronic"

Electronic is a branch of science and technology that deals with the behavior of electrons and it also means application of this, especially in developing equipment.

2.1.2 Definition of "Medical"

The meaning of medical is pertaining to medicine or to the treatment of diseases, pertaining to medicine as opposed to surgery. From Oxford Advanced Learner's Dictionary (Sixth Edition, 2000), the term "medical" connected with illness and injury and their treatment.

2.1.3 Definition of "Record"

Record is a written account of something that is kept, so that it can be looked at and used in the future. As in computing, record is defined as a collection of related fields within a single entity. It is also a collection of fields arrange in a predefined format.

2.2 Manual-based Medical Records System

In general, there are many hospitals and clinics still using method to manage medical center's information such as patients and medical records. All patient information, medication information and treatment involved huge amount of paper documents. In the traditional way, patient's profiles and medical records are write in the file and stored in the cabinet.

When the patient visits the medical center, or come for consultation, they have to show their registration card or identity card to the receptionist. Then, the staff will search the patient's file in the cabinet based on the registration number or their name if they forget to bring along the registration card. After the patient's record is found, patient will be given one queuing number. The file then will be passed to the physician in the room. Physicians traditionally record treatment and prescription as hand written progress notes in files. Hand written information is neither structured nor codes, and therefore cannot be easily used for automated decision support, research and outcomes analysis. Then,

patient's file will be passed to the dispensary department for giving medicine. The overall operating of a group of clinics in a hospital consumes a considerable of time from registration, searching for patient's record, until to pharmacists.

2.3 Computer-based Medical Records System

Due to inefficiency operating using manual system, many medical centers are changing their system to computerized system to assist their daily operation. All information or records are saver in computer. Therefore, user can easily retrieve and manage all related information such as medical records, medication information and others.

When patients come to medical center for treatment, they need to show their identity card to the receptionist. The receptionist will input their identity number or name into the system and easily found their record. The patient's records will be shown in the screen of the computer. This would certainly save a lot of time searching for patient's records compared to the manually operated system. Once the record is displayed, a queuing number will be generated to the patient and put into waiting list. If a new patient comes, registration is carried out. All the data are input through a standard form and a new record is carried out. All the data are input through a standard form and a new record is created for that particular patient. Also, the record of the patient is displayed on the computer screen in the physician's room. All the treatment, diagnostic and prescription of the medicine to the part can key in through the keyboard and save in the database easily.

Prescriptions are forwarded to the pharmacist. Pharmacists can base on the information displayed and gives the appropriate medicine to the patient. This will indeed reduce errors, potential risks and patients' waiting time.

The management of the medications and other clinic utilities also benefits from the computerized system. The medications inventory will automatically update after the medicine is giving away to the patient. Pharmacist can know the quantity of the medications by just clicking a few keyboard buttons rather than to check the remaining one-by-one. The staff only can access the inventory of the clinic utilities in the database. Message will be show when quantity of any clinic utilities below than certain amount. After that, staff in charge will start the procurement for that clinic utility.

2.4 Comparisons between manual and computer-based system

Below is the comparisons between manual and computer based system:

Table 2.1: Comparisons between manual and computer based system

Manual Medical Records System	Computer-based medical Records System
Records are written in papers and stored in cabinet or cupboard	Records are stored in database
Staff needs to search patient's record one-by-one in the cabinet, it may take time	Records can be search easily through database system
Risks lost or misplaced records are high	Records are stored safely and securely in the database
Hand-written may be not structured, it is difficult to read or analyze	All information is keep into standard form and easily to refer
Inventory of medications and other clinic utilities keep in a book. It is difficult to update the quantity of the medications	All inventory of medications and clinic utilities stored into system. Staff easily to notice the quantity of the medications
Analysis of patient's ratio, calculating of total income, reports for medication	Analysis of patient's ratio, calculating of total incomes, reports for medications and etc are easily to be generated
The management way is based on	The management is based on system and

individual. So, if a new staff quits, new staff needs a lot of time to pick up what the previous staff did. He or she also may be performed the management by his or her way. It is may cause the inconsistent	not individual, so it is easy for new staff to learn the system and it is more formal

2.5 Review of existing systems

2.5.1 Hospital Selayang

http://www.selayanghospital.gov.my

This website provides information to the user. This website was design and constructed for a TOTAL HOSPITAL INFORMATION SYSTEM (T.H.I.S) environment with ultimate aim of paper-less and firm-less hospital operation. It is the first hospital in Malaysia and the world to operate with THIS to cover all aspect of its operation. In order to meet the objective of the state of the art facility, a highly quality effective organization and management has to be ensured for the success of this hospital. THIS covers all aspects of its operation that consists 4 main modules:

- Patient management system
- Clinical System
- Clinical support system
- Administrate hospital system

As a summary, patients' medical records guidelines and clinical protocols are instantly available and can be assessed in one integrated workstation at any place and at anytime in the hospital, provided that the user has proper and rightful authority to access the information. Being an electronic hospital that has taken the initiative to implement THIS, Selayang Hospital has become a showcase to the rest of the world.

Figure 2.1: Hospital Selayang's homepage

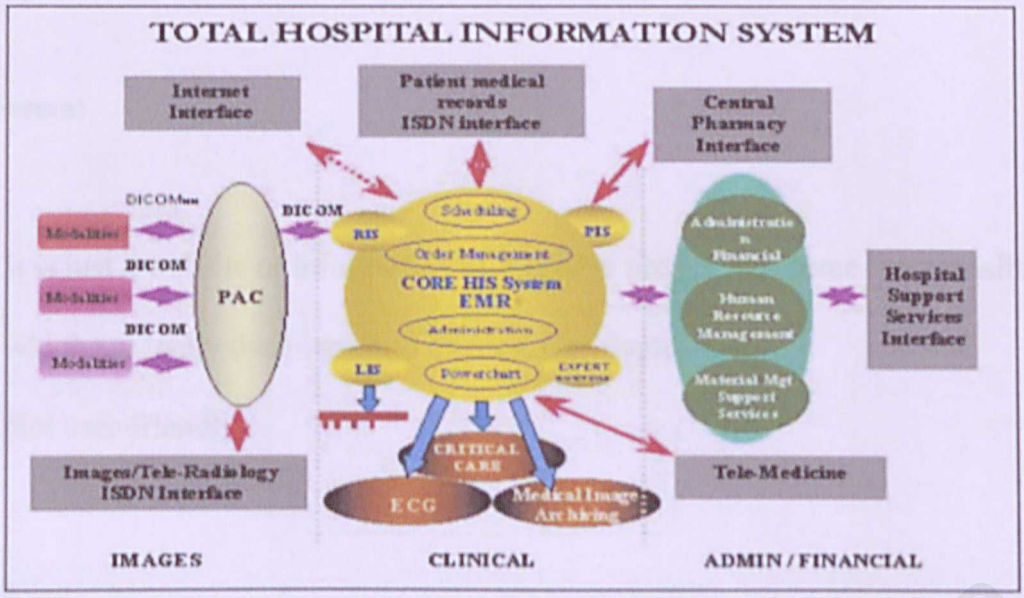


Figure 2.2: Total Hospital Information System (T.H.I.S)

Strength:

- Patient's medical records guidelines and clinical protocols are instantly available and can be accessed in one integrated workstation at any place and at anytime in the hospital
- It also provide a Flash site for an animated view
- Its design is simple but effective
- Information layout and the graphic images used are evenly balanced
- Some of the services that are very useful are hospital information and appointment form

Weaknesses:

- It is just a website of information as it is does not provide some functionalities which are really useful such as medical certificate
- Not user-friendly

2.5.2 Kajang Medical Centre

http://www.kajangmc.com

This website also provide information to the user. Their services are such as patient register, teaching and surgeons forum and also the name list of doctors. In past ten years Kajang Medical Centre has being one of the best hospitals in general orthopedics

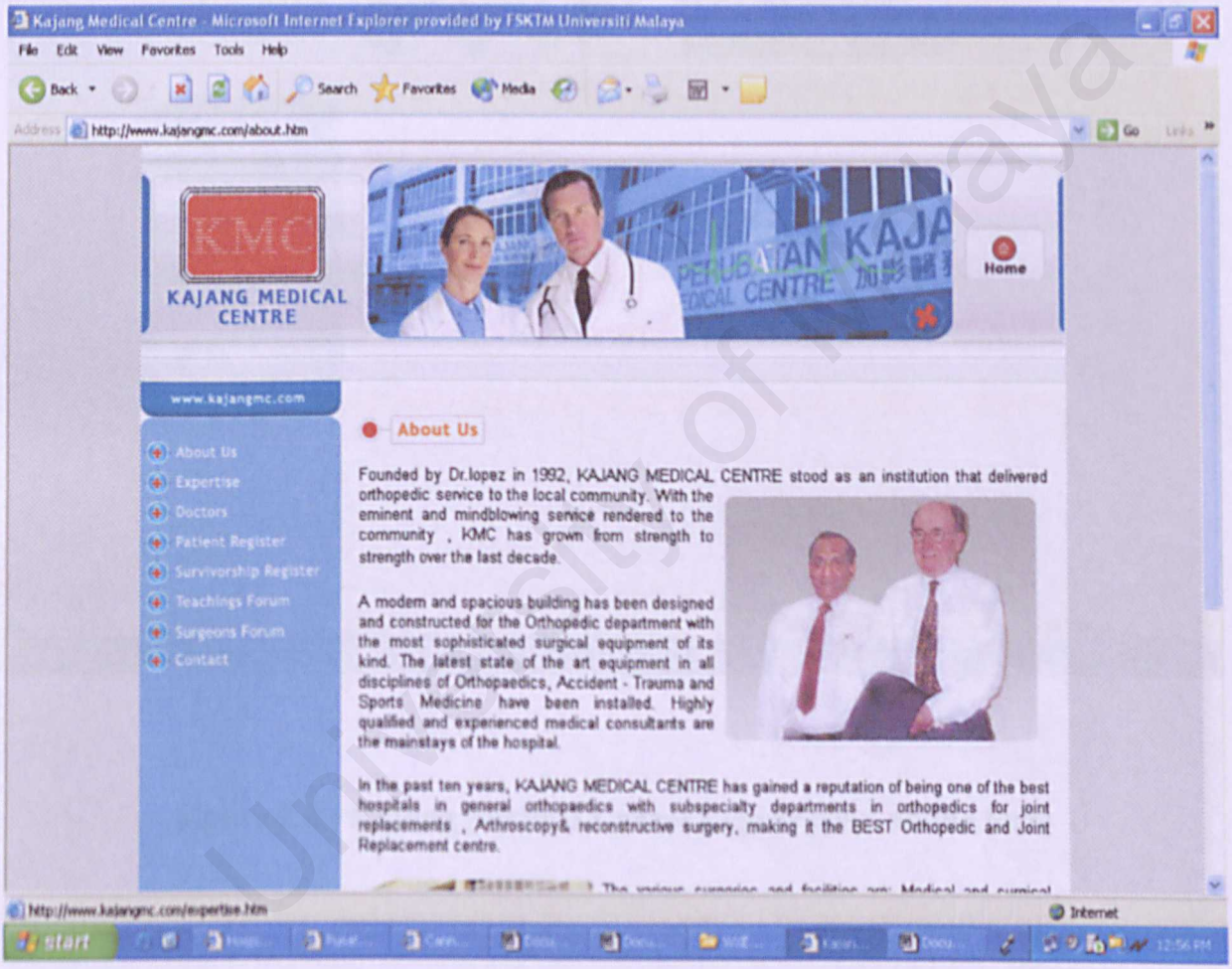


Figure 2.3 : Kajang Medical Centre Homepage

Patient registration form :

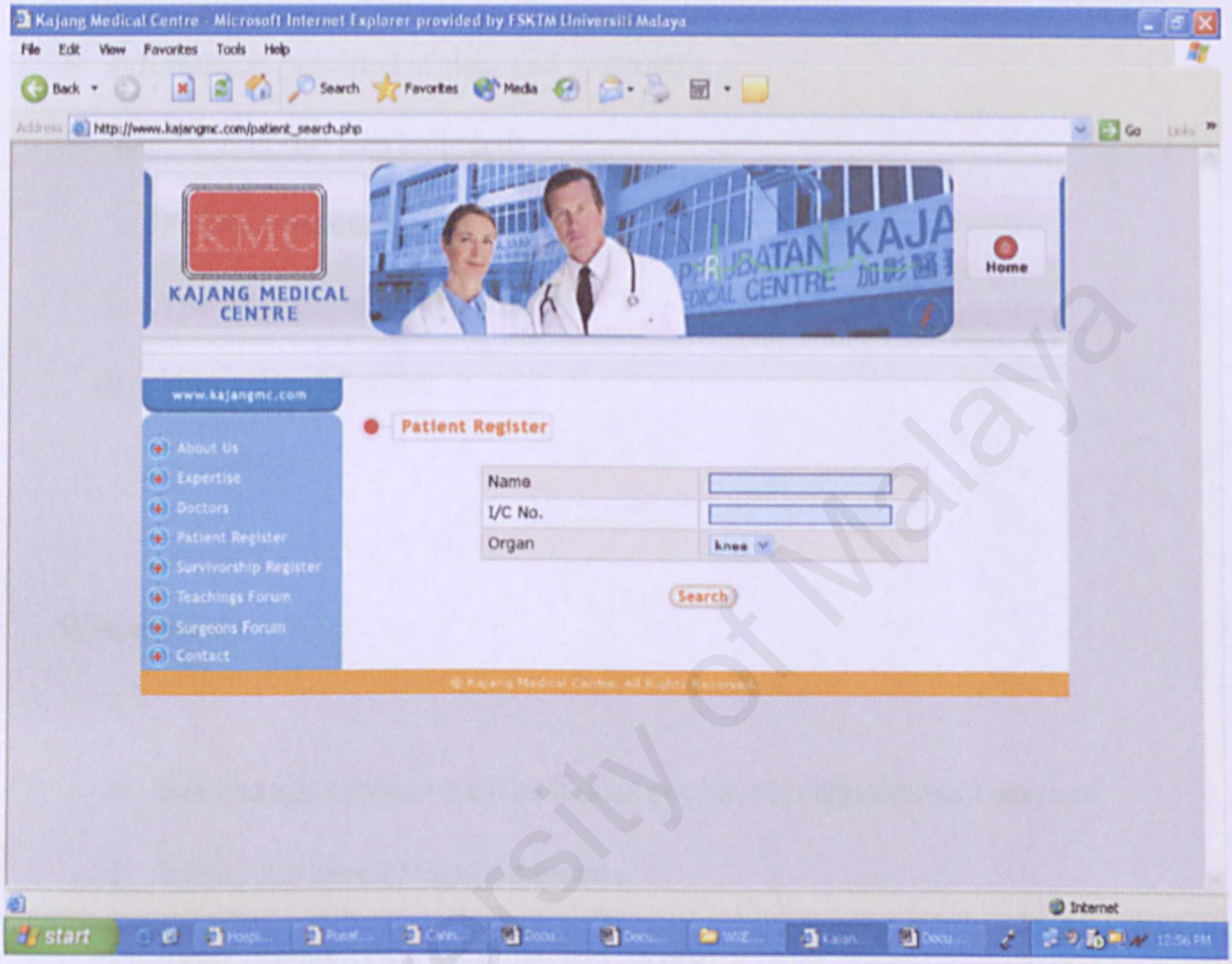


Figure 2.4 : Kajang Medical Centre Registration online form

Strength:

- This website is created using simple design
- The use of colors, graphic images and fonts are well coordinated and appropriate
- Information provided is clear and systematic
- Their useful services included:

I. Patient registration

II. Teaching and surgeons forum

III. Name list of doctors

Weaknesses:

- Not enough services such as it does not have an appointment services
- It does not have navigation menu
- Patient cannot view their medical record
- Not user friendly

2.5.3 TeleMedical.Com

http://www.telemedical.com

This website empowers individual to better manage their health and use an internet appliance for their medical needs. Telemedical.com's products and services are designed to increase access, decrease the cost and improve the quality of healthcare. Telemedical.com provides services such as :

- Digital media consulting
- Product shopping
- Medical video visits
- Clinical trials management
- Homecare extranet

Strength:

- Attractive and colorful site
- Interface are easy to use and user-friendly
- Presentation of the information is systematic

Weaknesses:

- Their services is hard to understand for the first-time user which some applications is more to the middle-user that have skills in IT field

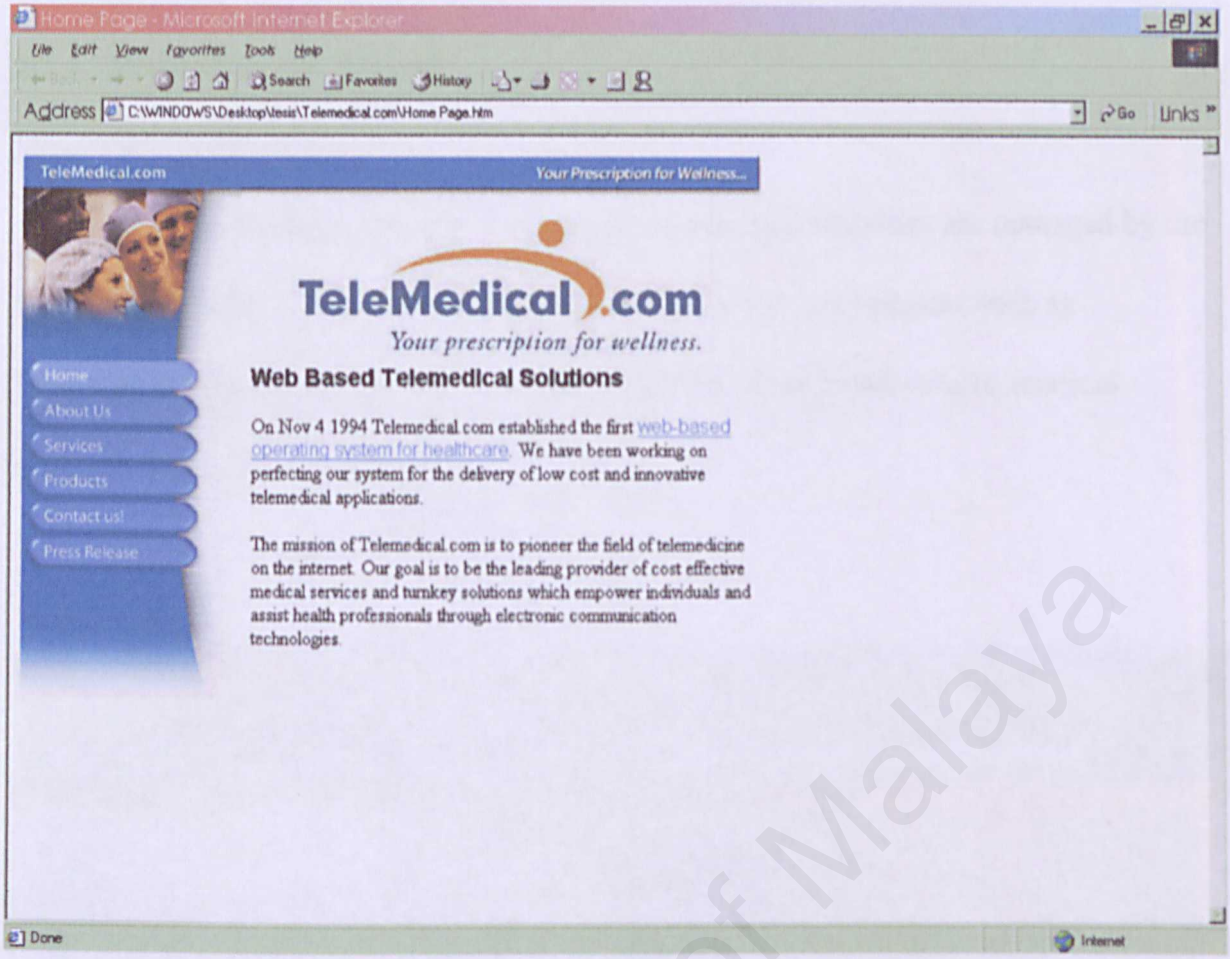


Figure 2.5 : Telemedical.com homepage

2.5.4 Personal Health Record

http://www.capmed.com

PHR medical information is organized on tabs, and activities are managed by the tab icons, tools bar controls and drop down menus. It services contains such as information medical records, visits and hospitalizations, medicals details, medical summary and communications.

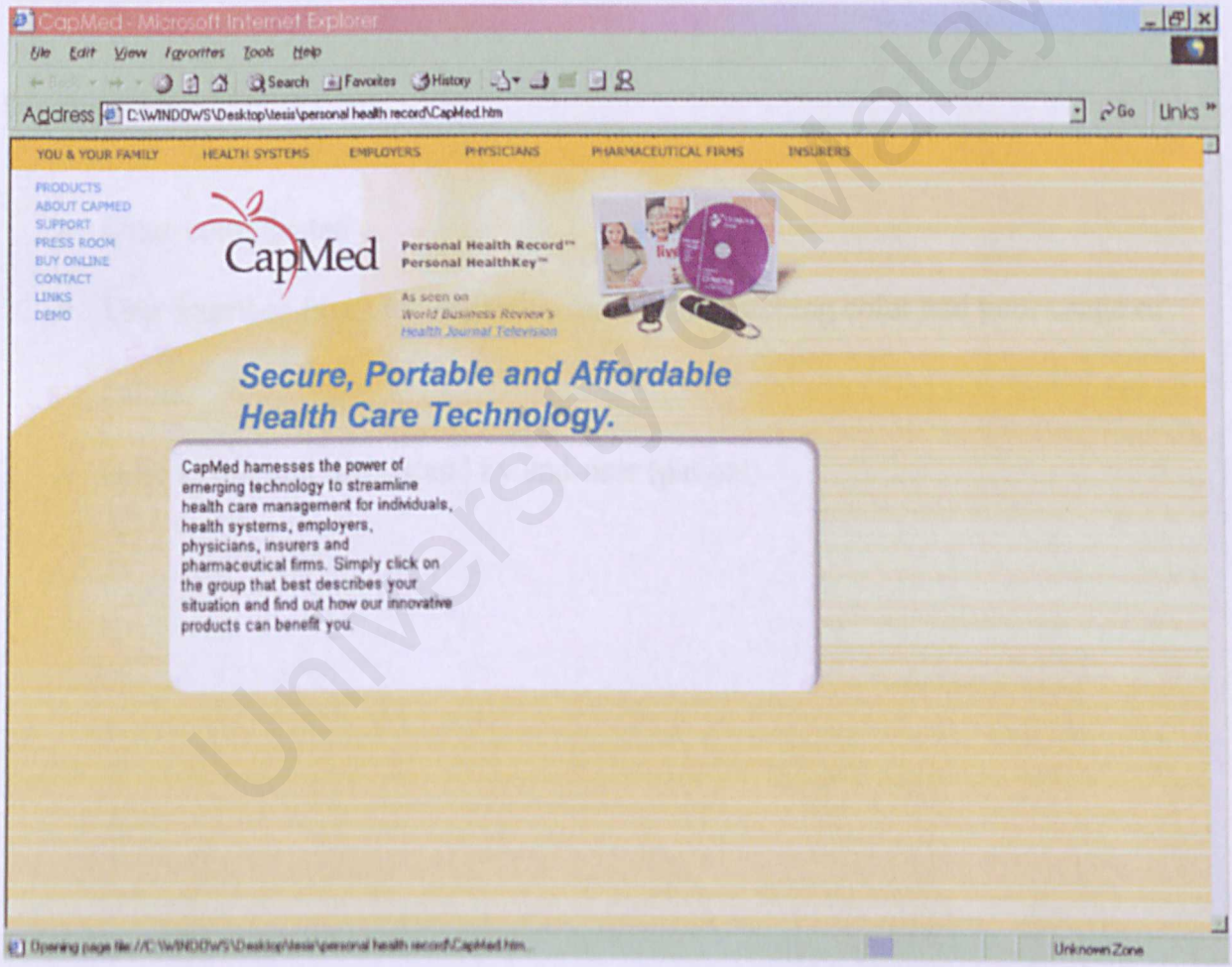


Figure 2.6 : Personal Health record Homepage

Strength:

- It provides information medical records, visits and hospitalizations, medical details, medical summary and communications
- Presentation of the information is systematic
- Interesting and dynamic contents
- For first time user, it also provide demonstration
- Well-structured system

Weaknesses:

- Quite complicated
- User interface is not too attractive due to mismatching color and poor usage of colors
- Hard to use and understand by end-user (patient)

2.5.5 Patient Centered Access to Secure System Online

http://www.pcasso.com

PCASSO's primary purpose is to enable health care providers and patient to access health information, including sensitive clinical data –securely over internet. To protect this sensitive information, PCASSO uses variety of security features beyond those used in electronic commerce applications such as credit card purchases. PCASSO is an information retrieval system, not a data entry system. So, it is not possible to enter information into the medical records or change the data.

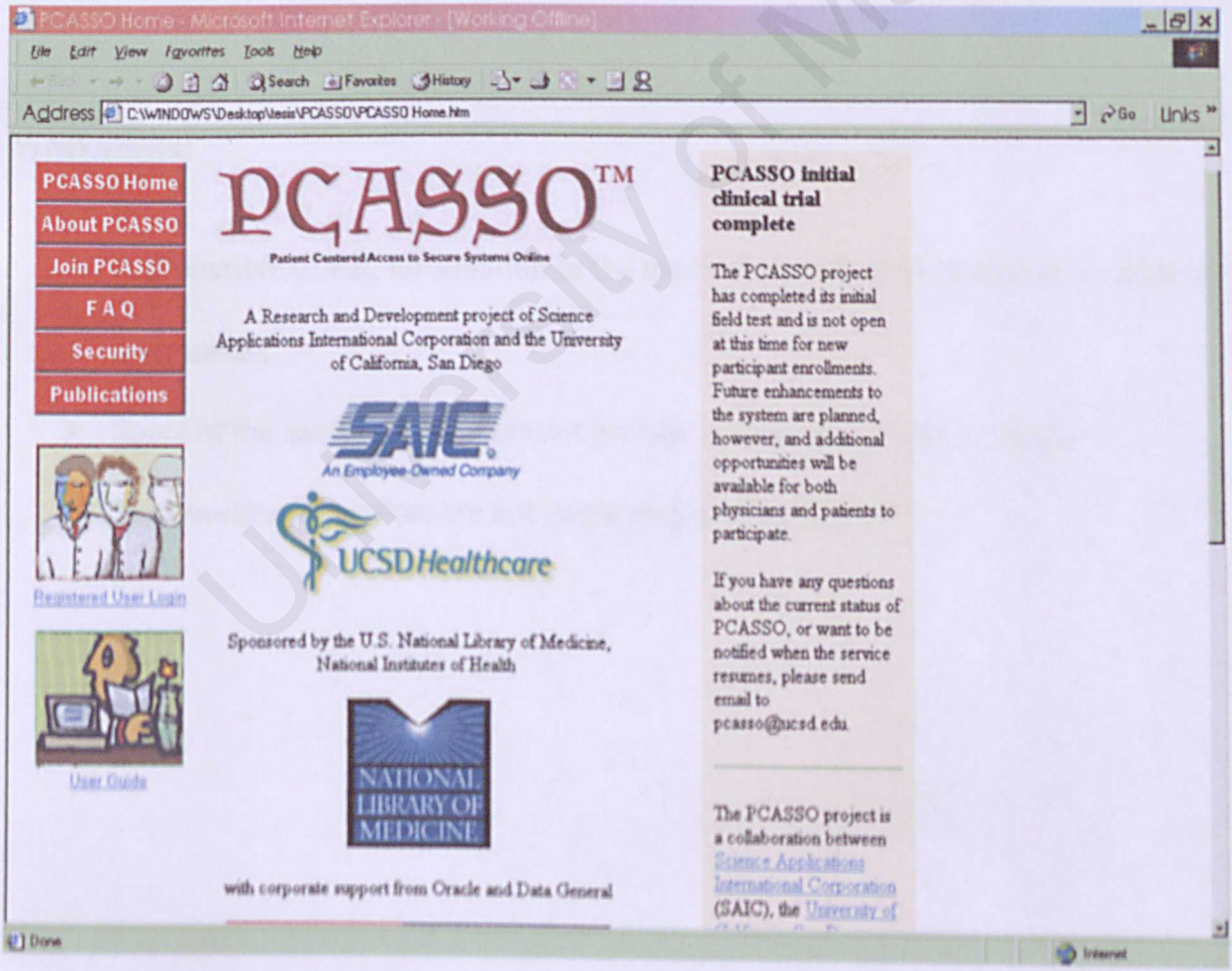


Figure 2.7 : Patient Centered Access to Secure System Online Homepage

Strength:

- Allows searching and display health information, provides demographics, lab tests and visit notes over the internet
- Encourage user's feedback
- Pleasant and smooth browsing
- Well-structured and systematic
- Interface are easy to use and user-friendly
- Provides online user guides and help pages

Weaknesses:

- It is possible to edit information in the medical record, or to change or to print the information
- Some of the useful services are not include such as appointment setting
- The coordination colors are not interesting and not attractive

2.5.6 Misys Electronic Medical Record

http://www.misys.com

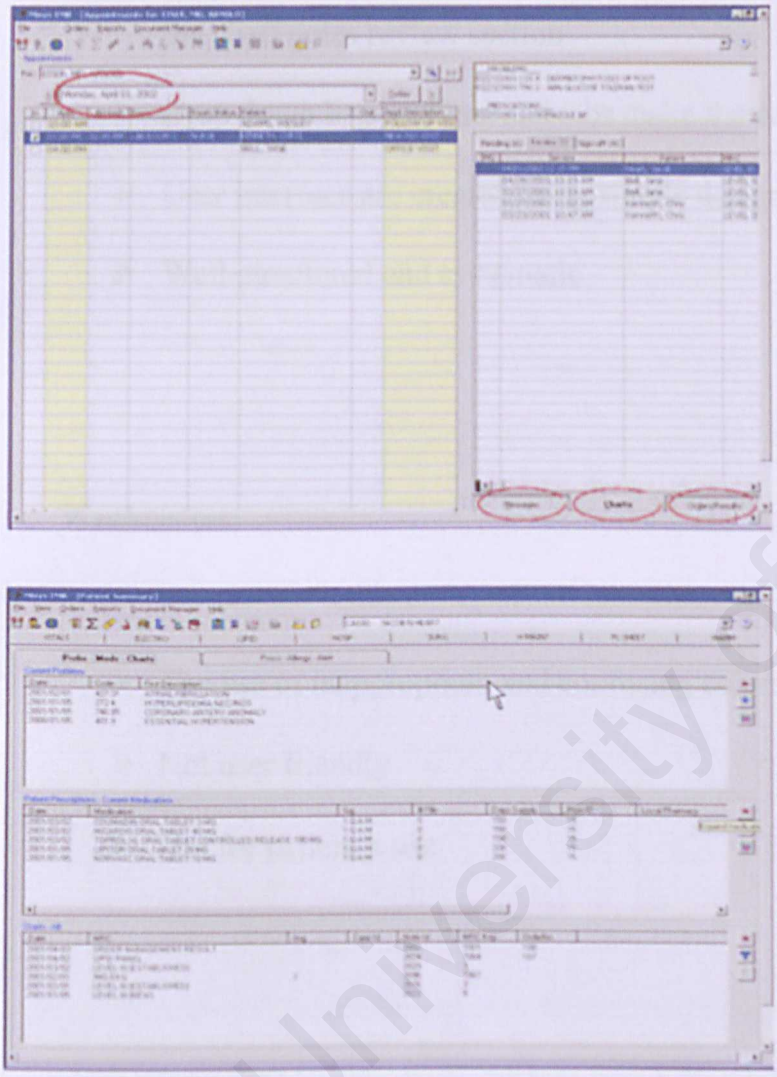


Figure 2.8 : Example of Misys.com system

Strength:

- Can review the daily schedule of appointments, e-mail, the orders review area and the chart notes review section
- Drop down lists, buttons and tabs make the navigation smooth
- User interface are created using simple design, so it is easy to understand
- Well-structured and systematic

Weaknesses:

- The use of inappropriate and confusing colors
- Not user friendly
- Not for patient's use

2.6 The .NET Framework

The .NET Framework is a Multilanguage environment for building, deploying and running Web Services and applications. It consists of two main parts:

- *Common Language Runtime*

  Despite its name, the runtime actually has a role in both a component's runtime and development time experiences. While the component is running, the runtime is responsible for managing memory allocation, starting up and killing threads and processes, and enforcing security policy, as well as satisfying any dependencies that the component might have other components. At development time, the runtime's role changes slightly; because it automates so much (for example, memory management), the runtime makes the developer's experience very simple, especially when compared to COM today. In particular, features such as reflection dramatically reduce the amount of code a developer must write in order to turn business logic into a reusable component.

- *Unified programming classes*

  The frameworks provide developers write a unified, object oriented, hierarchical, extensible set of class libraries (APIs). Currently, C++ developers use the Microsoft Foundation Classes. Java developers use the Windows Foundation Classes and Visual basic developers use Visual basic's APIs. The framework unifies these disparate frameworks. By creating a common set of APIs across all programming languages, the .NET framework enables cross-language inheritance, error handling and

debugging. All programming language, from Jscript to C++, become equals and developers are free to choose the language that they want to use.

2.7 Client/Server Architecture

The term client/server was first used in the 1980s in reference to personal computers (PCs) on a network. The actual client/server model started gaining acceptance in the late 1980s. The client/server software architecture is a versatile, message-based and modular infrastructure that is intended to improve usability, flexibility, interoperability and scalability as compared to centralized, mainframe and time sharing computing.

Client/server describes the relationship between two computer programs in which one program, the client, makes a service request from another program, the server, which fulfills the request. Although programs within a single computer can use the client/server idea, it is a more important idea in a network. In a network, the client/server model provides a convenient way to interconnect programs that are distributed efficiently across different locations. Computer transactions using the client/server model are very common. For example, to check your bank account from your computer, a client program in your computer forwards your request to a server program at the bank. That program may in turn forward the request to its own client program that sends a request to a database server at another bank computer to retrieve your account balance. The balance is returned back to the bank data client, which in turn serves it back to the client in your personal computer, which displays the information for you.

In the usual client/server model, one server, sometimes called a daemon, is activated and awaits client requests. Typically, multiple client programs share the services of a common server program. Both client programs and server programs are often part of a larger program or application. Relative to the internet, your Web browser is a client program that requests services (the sending of Web pages or files) from a web server in another computer somewhere on the internet. Similarly, your computer with TCP/IP installed you to make client requests for files from File Transfer Protocol (FTP) servers in other computers on the internet.

A client is defined as a requester of services and a server is defined as the provider of services. A single machine can be both a client and a server depending on the software configuration.

As a result of the limitations of file sharing architectures, the client/server architectures emerged. This approach introduced a database server to replace the file server. Using a relational database management system (DBMS), user queries could be answered directly. The client/server architecture reduces network traffic by providing a query response rather than total file transfer. It improves multi-user updating through a GUI front end to a shared database. In client/server architectures, Remote Procedure Calls (RPCs) or standard query language (SQL) statements are typically used to communicate between the client and server.

2.7.1 Two Tier Architectures

Two tier software architectures were developed in the 1980s from the file server software architecture design. The two tier architecture is intended to improved usability by supporting a forms-based, user friendly interface. The two tier architecture improves scalability by accommodating up to 100 users (file server architectures only accommodate a dozen users), and improves flexibility by allowing data to be shared, usually within a homogeneous environment. The two tier architecture requires minimal operator intervention, and is frequently used in non-complex, non-time critical information processing systems.

Two tier architectures consist of three components, distributed in two layers: client (requester of services) and server (provider of services). The three components are user system interface (such as session, text input, dialog, and display management services), processing management (such as process development, process enactment, process monitoring, and process resource services), and database management (such as data and file services). The two tier design allocates the user system interface exclusively to the client. It places database management on the server and splits the processing management between client and server; creating two layers. Figure 2.1 depicts the two tier software architecture.

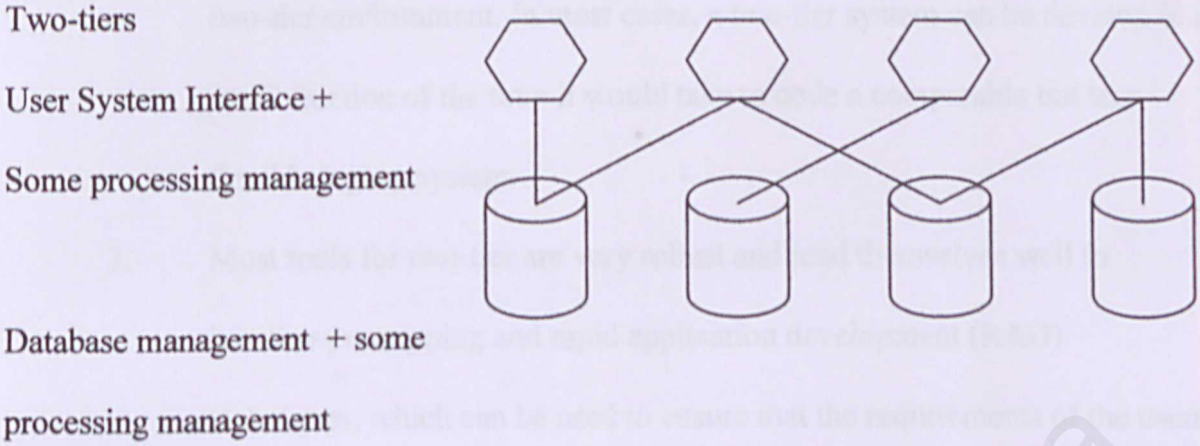


Figure 2.9: Two-tier client Server architecture design

In general, the user system interface client invokes services from the database management system server. In many two-tier designs, most of the application portion of processing is in the client environment. The database management server usually provides the portion of the processing related to accessing data (often implemented in store procedures) client commonly communicate with the server through SQL statements or a call-level interface. It should be noted that connectivity between tiers could be dynamically change depending upon the user's request for data and services.

2.7.1.1 Advantages of two-tier system

1. Application development speed is the most compelling advantage of a two-tier environment. In most cases, a two-tier system can be develop in a small fraction of the time it would take to code a comparable but less-flexible legacy system.
2. Most tools for two-tier are very robust and lend themselves well to iterative prototyping and rapid application development (RAD) techniques, which can be used to ensure that the requirements of the users are accurately and completely met.
3. Two-tier architectures work well in relatively homogenous environments with fairly static business rules. They are less suitable for dispersed, heterogeneous environment with rapidly changing rules.

2.7.1.2 Disadvantages of two-tier System

1. Because the bulk of application logic exists on the PC client, the two-tier architecture faces several potential version control and application redistribution problems. A change in business rules would require a change to the client logic in each application in a corporation's portfolio affected by the change.
2. System security in the two-tier environment can be complicated because a user may require a separate password for each SQL server accessed. The proliferation of end-user query tools can also compromise database server security.

3. Client tools and the SQL middleware used in two-tier environments are also highly proprietary, and the PC tools market is extremely volatile. The volatility of the client/server tool market raises questions about the long term viability of any proprietary tool an organization may commit to and complicates implementation of two-tier systems.

2.7.2 Three Tier Architecture

The three tier software architecture (three layer architectures) emerged in the 1990s to overcome the limitations of the two tier architecture. The third tier (middle tier server) is between the user interface (client) and the data management (server) components. This middle tier provides process management where business logic and rules are executed and can accommodate hundred of users (as compared to only 100 users with two tier architecture) by providing functions such as queuing, application execution and database staging. The third tier architecture is used when an effective distributed client/server design is needed that provides (when compared to two tier) increased performance, flexibility, maintainability, reusability and scalability, while handling the complexity of distributed processing from the user.

A three tier distributed client/server architecture (as shown in figure 2.2) includes a user system interface top tier when user services (such as session, text input, dialog and display management) reside.

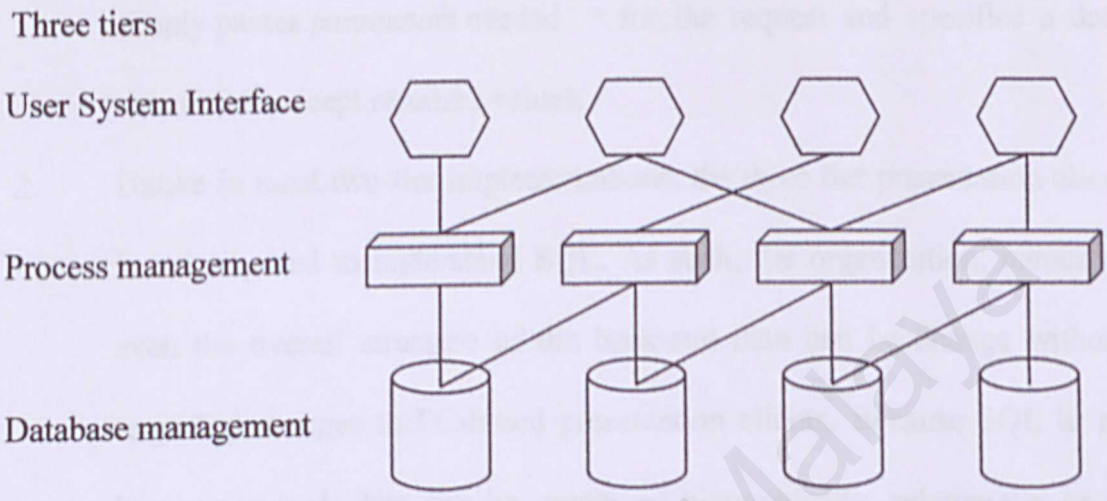


Figure 2.10: three tier distributed client/server architecture depiction

The third tier provides database management functionality and is dedicated to date and file services that can be optimized without using any proprietary database management system language. The data management component ensures that the data is consistent throughout the distributed environment through the use of features such as data locking, consistency and replication. It should be noted that connectivity between tiers could be dynamically change depending upon the user's request for data and services.

2.7.2.1 Advantages of Three-tier Architecture

1. RPC calls from presentation client to middle-tier server provide greater overall system flexibility than the SQL calls made by clients in the two-tier architecture. This is because in an RPC, the requesting client simply passes parameters needed for the request and specifies a data structure to accept returned values.
2. Unlike in most two-tier implementations, the three tier presentation client is not required to understand SQL. As such, the organization, names or even the overall structure of the back-end data can be change without requiring changes to PC-based presentation clients. Because SQL in no longer required, data can be organized hierarchically, relationally or in object format. This added flexibility allows a firm to access legacy data and simplifies the introduction of new database technologies.
3. Having separate software entities allows for the parallel development of individual tiers by application specialists.
4. Provides for more flexible resource allocation. Middle-tier functionality servers are highly portable and can be dynamically allocated and shifted as the needs of the organization change. Network traffic may be reduced by having functionality servers strip data to the precise structure required before distributing it to individual clients at the local area network (LAN) level.
5. Modularly designed middle-tier code modules can be reused by several applications. Reusable logic reduces subsequent development efforts,

minimizes the maintenance workload and decreases migration costs when switching client applications.

6. Three tier systems such as Open Software Foundation's Distributed Computing Environment (OSF/DCE) offer a variety of additional features to support distributed applications development.

2.7.2.2 Disadvantages of Three-tier Architecture

1. Three-tier brings with it an increased need for network traffic management, server load balancing and fault tolerance.
2. Current tools are relatively immature and require more complex 3GLs for middle-tier server generation. Maintenance tools have underdeveloped facilities for maintaining server libraries a potential obstacle for simplifying maintenance and promoting code reuse throughout an IS organization.

2.8 Summary of chapter 2

Overall, this chapter consist of 3 main parts; which are definitions, manual system vs computer based system and existing systems. For the existing system its give the explanation on some of the related system that have existed and how they perform their system. The definition of the project title also important to make sure that we understand our system usability and its requirements.

CHAPTER 3 : METHODOLOGY

3.0 Introduction

After analyzing the survey and findings from the literature review in the previous chapter, this chapter will specify the justifications for the chosen methodology for the project, which also include all outlined procedures that should be covered in order for user to understand the project requirements better.

3.1 Fact-finding Techniques

The various resources that were surveyed can be summarized as fact-finding techniques in order to obtain and gather reliable, relevant, adequate and comprehensive information. Several fact-finding techniques are used consecutively to ensure the success in conducting the system development and appropriate planning. A few techniques have been used in this project to gather information. It is also called information gathering or data collection.

- **Surfing the internet**

  Internet surfing today's world is a very efficient way of gathering information. There are many web site available that provide useful and expertise information, which is needed in the system. Some web site had provided very useful samples that can serve as guidelines in determining system requirement.

- **Reading**

  A lot of published literatures have been read in order to gather information of the users' needs, system development needs and technical issues of the proposed system. All these can be categorized into the printed material (especially books and journals) and non-printed material such as electronic documents and so. Through reading, ideas are managed to get from books, magazines and journal to be implemented in the proposed electronic medical records. Useful information has been found from the computing magazines and newspaper.

- **Library FSKTM**

  Previous seniors' thesis have been read through in order to gain an overall understanding on how a system was developed, what were the functional and non-functional requirements, and other related data. The general structure of each thesis has also been observed to find out the steps taken in carrying out a thesis.

- **Group discussion and Brainstorming session**

  Brainstorming session among group member, course-mates and supervisor, is a productive way in collecting ideas, suggestions and solutions of this project. However advices from supervisor and opinion from partner are also very significant in order to make this project successful.

- **Interview**

  In order to well suit the system with the real world environment, interview is the one important portion to be carried out. By interviewing, actual information can be obtained.

- **Observations**

  The existing search system of the Electronic Medical Records had been reviewed. Through this technique, the method of managing medical record in the existing system has been observed and defined. Besides the existing system is test to find out the functionality and the problem faced by the system itself.

3.2 System Development Model

Software engineering is important in organizing and executing the development of a system. Systematic analysis using *System Development Life Cycle* (SDLC) is a standard methodology to ensure the development process fulfils all the required aspects. The proper sequence of developing a system is called a "life cycle" that is implemented to ensure all individuals involved or interested to be involved in the project has the knowledge on how the system will be developed. This methodology is proven effective and it is still being used at the moment.

The development of a system using SDLC is divided into several software process models such as the Waterfall Model, V Model, Prototyping Model, Transformational Model and other models. This Medical Record System will be developed using the *Waterfall Model*. This model has several advantages over other models that ensure the development process is properly organized, high in quality and meets the standard.

3.2.1 The Waterfall Model

The system development is done based on the 5 major phases to make the system more organized and achieve its objectives. The phases are:

- Early review phase
- System analysis phase
- System design phase
- Coding/Programming phase
- Testing and execution phase

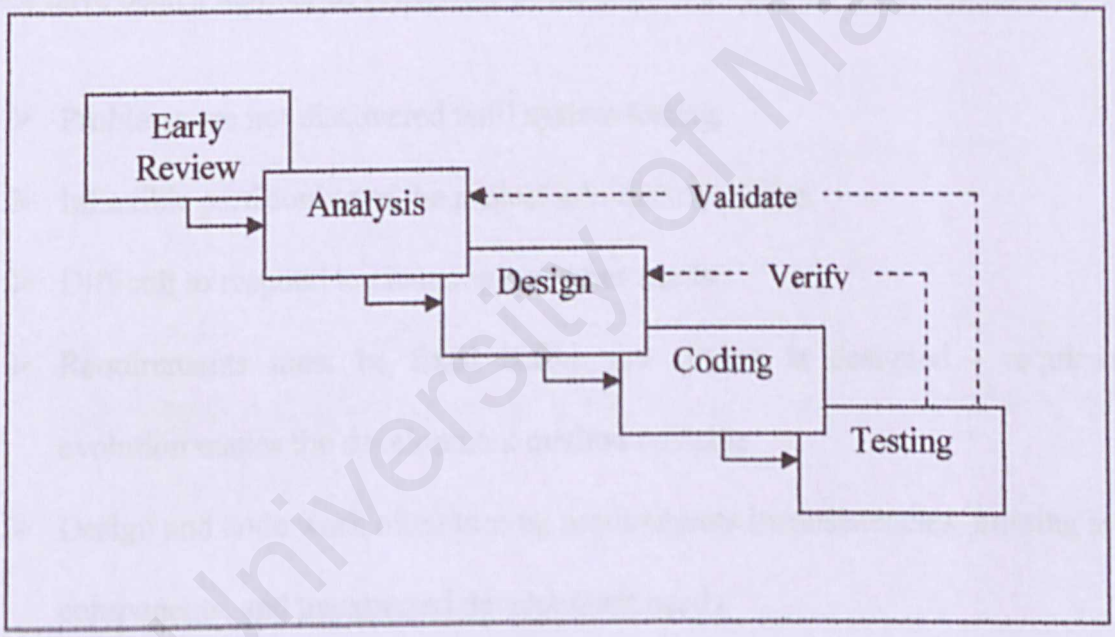


Figure 3.1: The implementation of Waterfall Model

The **advantages** of implementing this model are:

- Important stages in the development process can easily be recognized.
- It is easy to separate one stage from another.

- Presents a very high-level view of what of what goes on during development.
- It is easy to estimate period of time needed for an activity or stage.
- Testing is inherent to every phase of the waterfall model.
- It is an enforced disciplined approach.
- It is documentation driven, that is, documentation is produced at every stage.
- Validation ensures that the system has implemented the entire requirements, so that each system function can be traced back to a particular requirement in the specification.
- Verification ensures that all functions work correctly.

There have been a number of **criticisms** of the standard waterfall model, including

- Problems are not discovered until system testing.
- Inflexible partitioning of the project into distinct stages.
- Difficult to respond to changing customer needs.
- Requirements must be fixed before the system is designed - requirements evolution makes the development method unstable.
- Design and code work often turn up requirements inconsistencies, missing system components, and unexpected development needs.
- System performance cannot be tested until the system is almost coded; under capacity may be difficult to correct.

3.2.2 Project Development Phases

As mentioned above, there are 5 stages or phases in the development of this system- Electronic Medical Records. A brief description for each phases stated as below:

Phase I: Early Review

- It is also known as feasibility study. It could also be described as investigation and identification phase.
- Identification and Investigation Phase
  - At this point, early review is being done to gather information and description on the system that is going to be developed.
  - This phase will give the definition and the objective of the Electronic Medical Record in replacing the existing manual system.
  - Early investigation on the manual system has been done to identify the process involved and the problem that arises. User's demand is also being considered in this phase.

Phase II: System Analysis

- The analysis phase is done to understand how the Electronic Medical Record that is going to be developed will be able to solve the problems that exist in the manual system that had been identified from early reviews.

- It includes system review, system requirements and specifications, suggestions for system's content, development tools, analysis and interface medium being used.

Phase III: System Design

- Design is the creative process that requires understanding and natural talent to transform the problem into a solution. It is being acquired by reviewing systems that already exist. The description of a solution is also called design.
- The phase involves explanation about the whole system that is going to be developed and the expected system performance. In this phase all of the system properties such as system architecture, database design, process design, and interface design are being explained. This is done to simplify the interface developing process of the system that is going to be developed.
- The phase is being explained in details in Chapter 5.

Phase IV: Coding/Programming

- Programming is one of the most important aspects in developing a system. It will determine whether the system manage to achieve its objectives.

This makes the development environment vital in completing the programming process.

- All analysis and reviews had been done in details to determine the most suitable environment in achieving the project objectives. Besides all that, three main aspects that being emphasis on is the control structures, algorithm and data structure.

Phase V: Testing and Maintenance

- Testing is done to ensure that the system functions as it was suppose to. It is done to detect faults in the system so that all the modules developed are free from errors and the system can response to request effectively.
- Testing is one of the most important elements to ensure whether the system developed will be able to fulfill users' request. High quality system will be able to handle any type of system testing. To achieve this, all specification, design and programs done during the system development stage will be reviewed and re-evaluate.

3.3 Summary of chapter 3

In principle, the result of each phase is one or more documents which approved ('signed off'). The following phase should not start until the previous phase has finished. In practice, these stages overlap and feed information to each other. During **design,** problems with requirements are identified, during **coding** design problems are found and so on. The software process is not a simple linear model but involves in a sequence of iterations of the development activities.

Because of the cost of producing and approving documents, iterations are costly and involve significant rework. Therefore, after a small number of iterations, it is normal to freeze parts of the development, such as the specification, and to continue with the later development stages. Problems are left for later resolution, ignored or are programmed around. This premature freezing of requirements may mean that the system won't do what user wants. It may also lead to badly structured system as design problems are circumvented by implementation tricks.

During the final life-cycle phase **(operation and maintenance)** the software is put into use. Errors and omissions in the original software requirements are discovered. Program and design errors emerge and the need for new functionality is identified. The system must therefore evolve to remain useful. Making these changes **(software maintenance)** may involve repeating some or all previous process stages.

There are pros and cons of using the Waterfall Model and therefore, this model should only be used when the requirements are well understood. However, the Waterfall Model reflects engineering practice and thoroughly consideration has been made beforehand. Consequently, software processes based on this approach are still used for software development, particularly when this is part of a larger system engineering project.

CHAPTER 4: SYSTEM ANALYSIS

4.0 Introduction

System analysis is the process of extracting the needs of a system and what the system must do to satisfy the user's requirements. The goal of this analysis is to first understand the domain of the problem and the system's responsibilities by understanding how the users use or will use the system. In this chapter, we will focus on the requirements analysis. The term 'requirement' is not used throughout the software industry in a consistent way. In some cases, a requirement is seen as a high-level, abstract statement of a service that the system should provide or a constraint on the system. At the other extreme, it is a detailed, mathematically formal definition of a system function.

4.1 System Requirement Analysis

System requirement analysis can be divided into 2 sub-categories:

- Functional requirement
- Non-functional requirement

4.1.1 Functional Requirement

Functional requirement are functions or system abilities. Functional requirement also explain the interaction between the system and its environments. These are statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations. In some cases, the functional requirements may also explicitly state what the system should not do.

- *Password function*

Users have to login every time they want to use the system. This function requires user to input login name and password. Accessed will only be given if the password is correct for the login name specified.

- *Update record function*

The update function enables users to choose weather to eliminate, add or change the records in the database. The function requires input for update process and the updated record will be displayed as a result.

- *Patient's information record function*

This function will retrieve all patient information. At the end, the function will function will display a message of confirmation that the information has been recorded.

- *Patient history check function*

This function will display all of the information that had been recorded before from the patient's record.

- *Billing function*

This function is being used to change fees to patients for treatments and medications received. The information will then be forwarded to charge information form.

- *Search function*

This function enables user to track down record for a specific patient by their account number or patient identification card number.

4.1.2 Non-Functional Requirements

Non-functional requirements, as the name suggests, are those requirements that are not directly concerned with the specific functions delivered by the system. They may relate to emergent system properties such as reliability, response time, and store occupancy. Alternatively, they may define constraints on the system such as the capabilities of I/O (input/output) devices and the data representations. Non-functional requirements are limitations where system must operate to eliminate the limitation. Non-functional requirements for this system are as follows:

- **Reliability**

This system is reliable and will not require a high maintenance cost if it is used according to the correct procedures.

- **Security**

There are security features while accepting input or while retrieving data such as password protection.

- **Effectiveness**

This means that input and output screens have a specific purpose in the system

- **Simplicity**

Screens and instructions are organized properly so that it is much easier for user to understand and use the system.

- **User-friendly Interface**

Interesting interface is a vital aspect needed to encourage user to use the system. It is also a user-friendly based to make it easy for user especially first-timer to use this system.

4.2 Software Requirement

4.2.1 Server Software requirement

- Extensible Markup Language (XML) 1.0
- Microsoft Access 2003 – Database repository
- Windows NT Server 4.0- Network Operating System
- Microsoft Windows XP Professional
- Microsoft Internet Explorer 5.0
- Macromedia Dream weaver MX
- Notepad

4.2.2 Client Software requirement

- Microsoft Windows XP
- Microsoft Internet Explorer 5.0

4.3 Hardware Specification

4.3.1 Server Hardware requirement

- Pentium III 400MHz computer and above
- 20 GB Hard Disk Space
- At least 128 MB RAM
- Network interface Card (NIC) and network connection with recommended bandwidth at 15 Mbps or more
- Other standard computer peripherals
- Standard Input/Output Devices

4.3.2 Client Hardware requirement

- A Pentium 3 with 750 MHz processor
- At least a 64 MB RAM
- Network connection through existing network configuration or modem
- Other standard computer peripherals

4.4 Technology Considerations

4.4.1 Microsoft Windows XP

As the latest operating system, Windows XP is inherently more reliable than Windows NT. In addition to this, any "vulnerabilities" that are found in the operating system is acted upon quickly by Microsoft, ensuring that Windows XP will remain as stable and secure operating system as possible. A number of "behind the scenes" improvements and enhancements also ensure that Windows XP is a significantly better operating system than earlier versions, such as Windows NT. One of the most notable improvements is in the systems resilience to loss of network - if the PC were to lose it's network connection, Windows XP is able to continue unaffected. This is largely down to the fact that a number of applications, such as Microsoft Office, are installed locally on the hard disk - on Windows NT these applications are accessed over the network.

4.4.2 Internet Explorer 6

Internet Explorer 6 is a set of core technologies in Microsoft® Windows XP system that provides enhanced privacy features and a flexible and reliable browsing experience for users of Windows XP, Windows Millennium Edition (Windows Me), Windows 2000, Windows 98, and Windows NT® 4.0 with Service Pack 6a or later.

Whether you are a home user browsing content or getting e-mail on the Web, an IT administrator deploying and maintaining a rich set of Windows

Internet technologies, or a Web developer creating Web content, Internet Explorer 6 gives you the freedom to experience the best of the Internet.

Features and Technologies at a Glance

Internet Explorer 6 includes many new and enhanced features that can simplify the daily tasks that you perform, while helping you to maintain the privacy of your personal information on the Web. Here is a quick look at some of the major features of Internet Explorer 6 that help to provide a flexible and reliable browsing experience on the Web.

Flexibility

With new, innovative browser capabilities and features such as Auto Image Resizing, Image Toolbar, Media Bar, and Print Preview, it's easy to manage, save, and print your pictures and other media from Web pages. Features such as Favorites, Auto Complete, History, and the Search Companion help you quickly find what you need online. And the customizable browsing layout makes it easy to change your layout so you can experience the Web the way you want.

Web Privacy

Internet Explorer 6 helps you manage your security and privacy preferences while on the Internet with tools that help you safeguard your family's browsing experience. Manage cookies to help control the personal information that Web sites collect about you, set different levels of security for different sites on the Web with Security Zones, and use Content Advisor to help block access to

objectionable content. These tools support the Platform for Privacy Preferences (P3P), a technology under development by the World Wide Web Consortium (W3C).

Reliability

Internet Explorer 6 helps deliver a more stable and more reliable web browsing experience. New fault collection services help to identify potential problems that need to be fixed in future updates to Windows Internet technologies.

4.4.3 The .NET Framework

The .NET Framework is a Multilanguage environment for building, deploying and running Web Services and applications. It consists of two main parts:

- **Common Language Runtime**

  Despite its name, the runtime actually has a role in both a component's runtime and development time experiences. While the component is running, the runtime is responsible for managing memory allocation, starting up and killing threads and processes, and enforcing security policy, as well as satisfying any dependencies that the component might have other components. At development time, the runtime's role changes slightly; because it automates so much (for example, memory management), the runtime makes the developer's experience very simple,

especially when compared to COM today. In particular, features such as reflection dramatically reduce the amount of code a developer must write in order to turn business logic into a reusable component.

- **Unified programming classes**

  The frameworks provide developers write a unified, object oriented, hierarchical, extensible set of class libraries (APIs). Currently, C++ developers use the Microsoft Foundation Classes. Java developers use the Windows Foundation Classes and Visual basic developers use Visual basic's APIs. The framework unifies these disparate frameworks. By creating a common set of APIs across all programming languages, the .NET framework enables cross-language inheritance, error handling and debugging. All programming language, from Jscript to C++, become equals and developers are free to choose the language that they want to use.

4.4.4 Windows NT Server 4.0

Windows NT was developed from the ground up by Microsoft as an entirely new operating system, based on 32-bit code that is free from the limitations of MS-DOS or 16-bit Windows. The Microsoft development team, led by Dave Cutler (a former designer of Digital's VAX/VMS operating system), design the Windows NT operating system with the following design goals:-

- Reliability
- Compatibility

- Scalability
- Performance
- Portability
- Security

Microsoft® Windows NT® Server 4.0 is a multipurpose server operating system. A multipurpose operating system integrates a variety of network services. The services it provides are designed to address customer requirements and are managed in a single way. The features of Windows NT Server 4.0 include:

Microsoft Management Console

Microsoft Management console, released with Windows NT Server's built-in Web server, Internet Information Server (IIS) 4.0, combines the administration tasks of IIS into one tool and allows administrators to create task-based consoles that can be delegated to the appropriate administrator. This provides administrators with one integrated utility for managing their applications and Web environment.

Windows 95 User Interface

The Microsoft Windows 95 operating system user interface has been integrated into Microsoft Windows NT Server 4.0.

Administrative Wizards

The Microsoft Windows 95 operating system user interface has been integrated into Microsoft Windows NT Server 4.0.

Network Monitor

A network diagnostic tool, the Network Monitor examines network traffic to and from the server at the packet-level. It can capture network traffic so you can analyze it later.

4.4.5 XML

XML is a markup language for documents containing structured information. Structured information contains both content (words, pictures, etc.) and some indication of what role that content plays (for example, content in a section heading has a different meaning from content in footnote, which means something different than content in a figure caption or content in a database table, etc.). Almost all documents have some structure. A markup language is a mechanism to identify structures in a document. The XML specification defines a standard way to add markup to documents

XML is an open standard for defining data elements on a web page and business documents. In contrast to HTML markup language, which defines how elements on a web page are displayed, XML defines the structured information those elements contain. XML is also a markup language that was originated from a subset of SGML together with HTML to use over the Internet via web browsers. XML documents can be viewed as containers for information. Within the primary container may be information and more nested containers, which themselves may contain information and more nested containers (Simon, 2000).

An XML document is a database only in the strictest sense of the term. That is, it is a collection of data. As a 'database' format, XML has some advantages [as it is a meta-data language] because of its self- describing and it can describe data in tree or graph structures. It also has some disadvantages as it is verbose and access to the data is slow due to parsing and text conversion (Bourett, 2002)

From the paragraph taken above, a simple deduction can be made that XML and its surrounding technologies constitute a simple database format, or rather like the DBMS format. Some of its advantages are the ability to provide data storage and retrieval in XML document format, data schemas via DTDs and XML schemas, query language via XQuery, XPath, XQL, XML -QL and so on, programming interfaces via DOM and SAX (Bourret, 2002). On the other hand, its advantages are lack of features that are vital in any production environment such as efficient data storage and retrieval, indexes, security, and transactions, data integrity, performance, multi-user accesses and lastly performing complex queries across multiple XML storage. Therefore, it has to be coexisting with RDBMS to complement each other.

XML vs. HTML

In HTML, both the tag semantics and the tag set are fixed. An <h1> is always a first level heading and the tag <ati.product.code> is meaningless. The W3C, in conjunction with browser vendors and the WWW community, is constantly working to extend the definition of HTML to allow new tags to keep

pace with changing technology and to bring variations in presentation (stylesheets) to the Web. However, these changes are always rigidly confined by what the browser vendors have implemented and by the fact that backward compatibility is paramount. And for people who want to disseminate information widely, features supported by only the latest releases of Netscape and Internet Explorer are not useful.

XML specifies neither semantics nor a tag set. In fact XML is really a meta-language for describing markup languages. In other words, XML provides a facility to define tags and the structural relationships between them. Since there's no predefined tag set, there can't be any preconceived semantics. All of the semantics of an XML document will either be defined by the applications that process them or by stylesheets

XML vs. SGML

XML is defined as an application profile of SGML. SGML is the Standard Generalized Markup Language defined by ISO 8879. SGML has been the standard, vendor-independent way to maintain repositories of structured documentation for more than a decade, but it is not well suited to serving documents over the web (for a number of technical reasons beyond the scope of this article). Defining XML as an application profile of SGML means that any fully conformant SGML system will be able to read XML documents. However, using and understanding XML documents does not require a system that is

capable of understanding the full generality of SGML. XML is, roughly speaking, a restricted form of SGML.

For technical purists, it's important to note that there may also be subtle differences between documents as understood by XML systems and those same documents as understood by SGML systems. In particular, treatment of white space immediately adjacent to tags may be different.

Why XML?

In order to appreciate XML, it is important to understand why it was created. XML was created so that richly structured documents could be used over the web. The only viable alternatives, HTML and SGML, are not practical for this purpose. HTML, as we've already discussed, comes bound with a set of semantics and does not provide arbitrary structure. SGML provides arbitrary structure, but is too difficult to implement just for a web browser. Full SGML systems solve large, complex problems that justify their expense. Viewing structured documents sent over the web rarely carries such justification.

This is not to say that XML can be expected to completely replace SGML. While XML is being designed to deliver structured content over the web, some of the very features it lacks to make this practical, make SGML a more satisfactory solution for the creation and long-time storage of complex documents. In many organizations, filtering SGML to XML will be the standard procedure for web delivery.

XML Development Goals

1. It shall be straightforward to use XML over the Internet. Users must be able to view XML documents as quickly and easily as HTML documents. In practice, this will only be possible when XML browsers are as robust and widely available as HTML browsers, but the principle remains.
2. XML shall support a wide variety of applications. XML should be beneficial to a wide variety of diverse applications: authoring, browsing, content analysis, etc. Although the initial focus is on serving structured documents over the web, it is not meant to narrowly define XML.
3. XML shall be compatible with SGML. Most of the people involved in the XML effort come from organizations that have a large, in some cases staggering, amount of material in SGML. XML was designed pragmatically, to be compatible with existing standards while solving the relatively new problem of sending richly structured documents over the web.
4. It shall be easy to write programs that process XML documents. The colloquial way of expressing this goal while the spec was being developed was that it ought to take about two weeks for a competent computer science graduate student to build a program that can process XML documents.
5. The number of optional features in XML is to be kept to an absolute minimum, ideally zero. Optional features inevitably raise compatibility problems when users want to share documents and sometimes lead to confusion and frustration.

6. XML documents should be human-legible and reasonably clear. If you don't have an XML browser and you've received a hunk of XML from somewhere, you ought to be able to look at it in your favorite text editor and actually figure out what the content means.
7. The XML design should be prepared quickly. Standards efforts are notoriously slow. XML was needed immediately and was developed as quickly as possible.
8. The design of XML shall be formal and concise. In many ways a corollary to rule 4, it essentially means that XML must be expressed in EBNF and must be amenable to modern compiler tools and techniques. There are a number of technical reasons why the SGML grammar cannot be expressed in EBNF. Writing a proper SGML parser requires handling a variety of rarely used and difficult to parse language features. XML does not.
9. XML documents shall be easy to create. Although there will eventually be sophisticated editors to create and edit XML content, they won't appear immediately. In the interim, it must be possible to create XML documents in other ways: directly in a text editor, with simple shell and Perl scripts, etc.
10. Terseness in XML markup is of minimal importance. Several SGML language features were designed to minimize the amount of typing required to manually key in SGML documents. These features are not supported in XML. From an abstract point of view, these documents are indistinguishable from their more fully specified forms, but supporting these features adds a considerable burden

to the SGML parser (or the person writing it, anyway). In addition, most modern editors offer better facilities to define shortcuts when entering text.

4.4.6 Microsoft Access 2003

Access 2003 provides a powerful set of tools that are sophisticated enough for professional developers, yet easy to learn for new users. Create or use powerful database solutions that make organizing, accessing, and sharing information easier than ever.

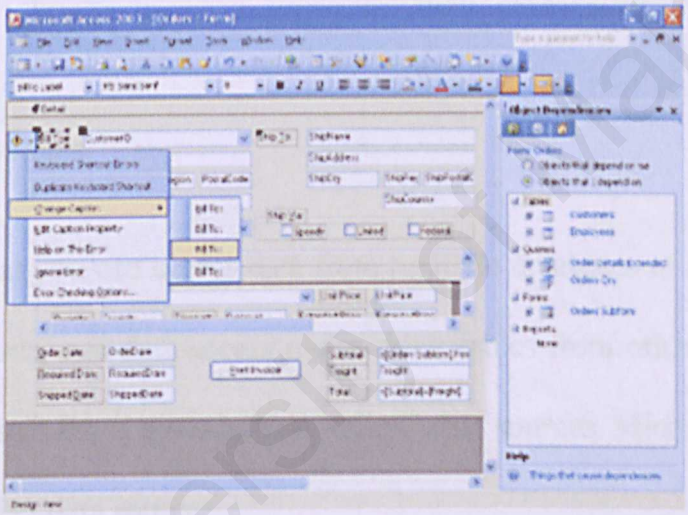


Figure 4.1: Microsoft Access 2003 workspace

Access and Use Information from Multiple Sources

Use information from a variety of formats and programs—all in a familiar interface.

- **Incorporate a wide range of data sources.**

  Access 2003 supports a variety of data formats, including Extensible Markup Language (XML), OLE, Open Database Connectivity (ODBC), and Microsoft Windows® SharePoint™ Services.

- **Link business systems.**

  Link tables so that you can access data from multiple databases at one time in Access 2003 forms, reports, and data-access pages. Link tables from other Microsoft Access databases, Microsoft Excel spreadsheets, ODBC data sources, Microsoft SQL Server™ databases, and other data sources.

- **Get the most from your enterprise data.**

  Incorporate SQL Server data into your Access 2003 solutions. Use the Stored Procedure Designer to create and modify simple procedures stored in SQL Server without having to learn Transact-SQL.

Depend on Access 2003

Reduce errors and get help when you need it.

- **Give your forms a new look.**

Access 2003 supports themes from Microsoft Windows XP to give your forms a consistent design.

- **View dependency information.**

Quickly find tables, queries, forms, or reports that depend on a particular database object.

- **Help eliminate errors.**

The new error checking feature flags common errors in forms and reports, making testing and fixing errors faster. After errors are flagged, you are given options to correct the errors, saving time and helping you to create more accurate forms and reports.

- **Update properties automatically.**

Change a field property in a table and all the forms or reports that have controls bound to it can be updated automatically.

- **Analyze your information in powerful ways.**

Drag controls onto your Access 2003 form to create a Microsoft PivotTable® or Microsoft PivotChart® view, or a spreadsheet.

- **Easily back up your information.**

  In Access 2003, save a copy of the database you're working on to another location.

- **Find the help you need.**

  From the Getting Started and Help task panes, access Assistance on Office Online. It provides help and assistance articles that are updated regularly from requests and issues of other users. Some functionality in these task panes requires a connection to the Internet.

Share Information with Confidence

Design forms and reports that you can save and open in previous versions of Access, making it easier to share your information.

- **Share information more efficiently.**

  Export and import data, and link to lists on Windows Share Point Services sites where other team members can access your data. Building these sites requires Microsoft Windows Server™ 2003 running Windows Share Point Services.

- **Design Web pages using powerful tools.**

  Publish forms and reports on the Web and bind your information to a record source to display, update, and work with data from your database.

- **Use compatible file formats.**

Access 2003 uses Access 2000 as the default file format for new databases. Because Access version 2002 and Access 2000 can use and modify the same database, organizations can roll out Access 2003 while maintaining capability with existing Access users and solutions.

4.5 Summary

In practice, it may be difficult to separate functional and non-functional considerations and to identify requirements which relate to the system as whole. In reality, the distinction between these different types of requirement is not as clear cut as these simple definitions suggest. It is also necessary to determine what kind of software or technology to be applied to fulfill the system's requirement. However, when developed in more detailed, it may lead to other requirement that is clearer and more relevant.

CHAPTER 5 SYSTEM DESIGN

5.0 Introduction

System design is the phase in which the requirements analyzed and produced in the system analysis phase are translated and converted into a representation characteristics of the proposed system. The design process involves developing several models of the system at different level of abstraction. As a design is composed, errors and omission in earlier stages are discovered. Generally, this phase will be focusing on architectural design, functional design, database design and the interface design.

5.1 Architectural Design

Large system can be decomposed into sub-system that provides some related set of services. The initial design process of identifying those sub-systems and establishing a framework for sub-system control and communication is called architectural design. It usually comes before the detailed system specification. (Figure 5.1)

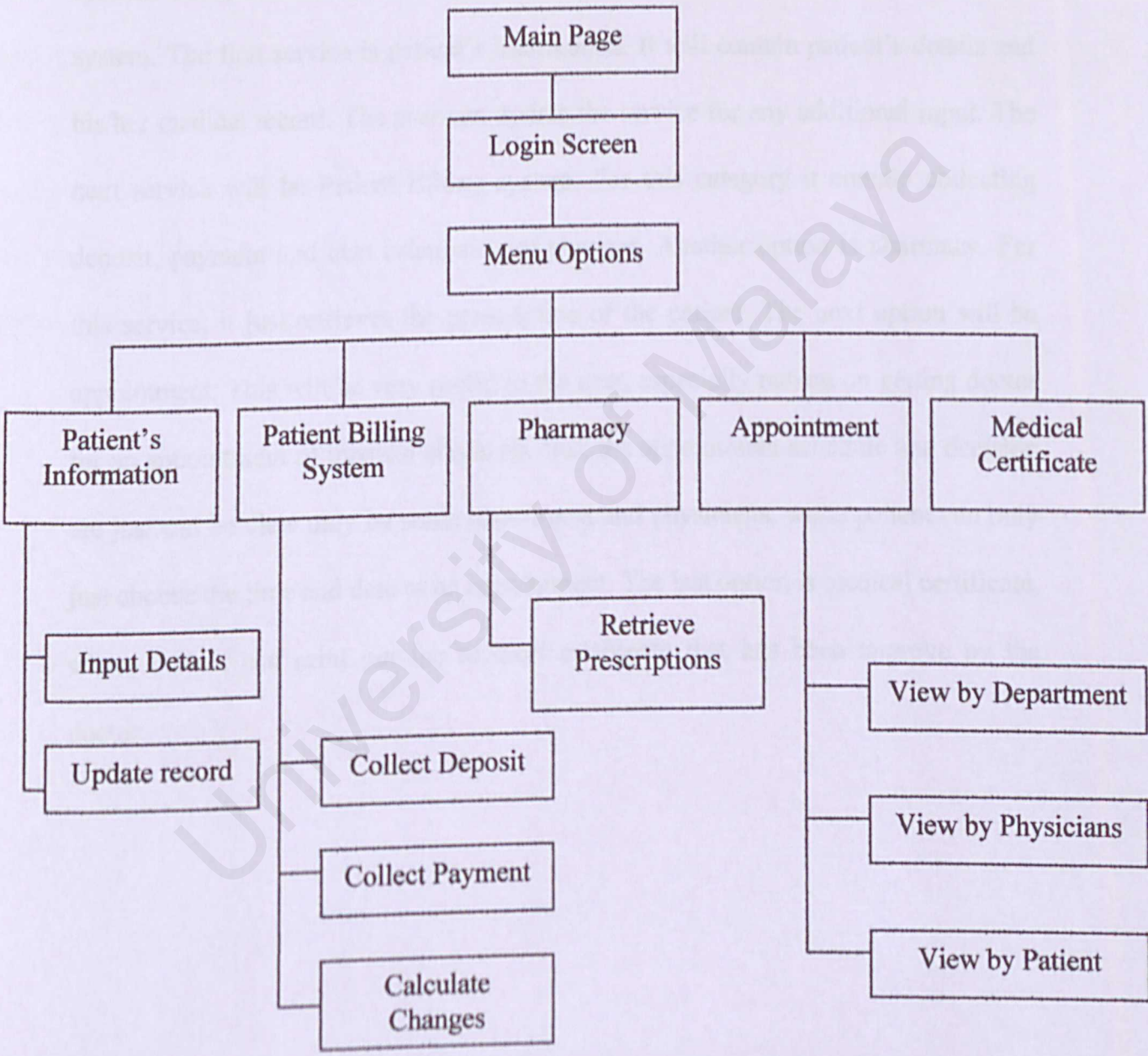


Figure 5.1 : Architectural View of Electronic Medical Record

5.1.1 Architectural design of Electronic Medical Record

The system will start at the main page, whereas everyone can view the system. But just for authorized users that have username and password are allow login the system. Then, there are several options that have to be choose by the user to use the system. The first service is patient's information. It will contain patient's details and his/her medical record. The user can update the service for any additional input. The next service will be Patient Billing system. For this category it contain collecting deposit, payment and also calculation of changes. Another option is pharmacy. For this service, it just retrieves the prescription of the patient. The next option will be appointment. This will be very useful to the user, especially patient on getting doctor for an appointment of medical check-up. But, the appointment schedule and decision are just can be view only by some department and physicians, while patient can only just choose the time and date of an appointment. The last option is medical certificate, where patient can print out the medical certificate that has been approve by the doctor.

5.2 Process Design

Process design will be visualize using Data Flow Diagram. It is a graphical technique that will display the data flow in the system. As a transformer of data, the diagram shows the data flow into the system, how they are transformed and how they leave the system. The emphasis is on the flow of the data, not on the flow of control. It will also be able to view the changing process or the converting process that is being implemented to the data once the data goes into the system, through the system and out of the system. The hierarchy is expressed by layering, so that different details are shown in different layers. It can be divided into several layers that can visualize the ascending data flow and functional information. (Figure 5.2)

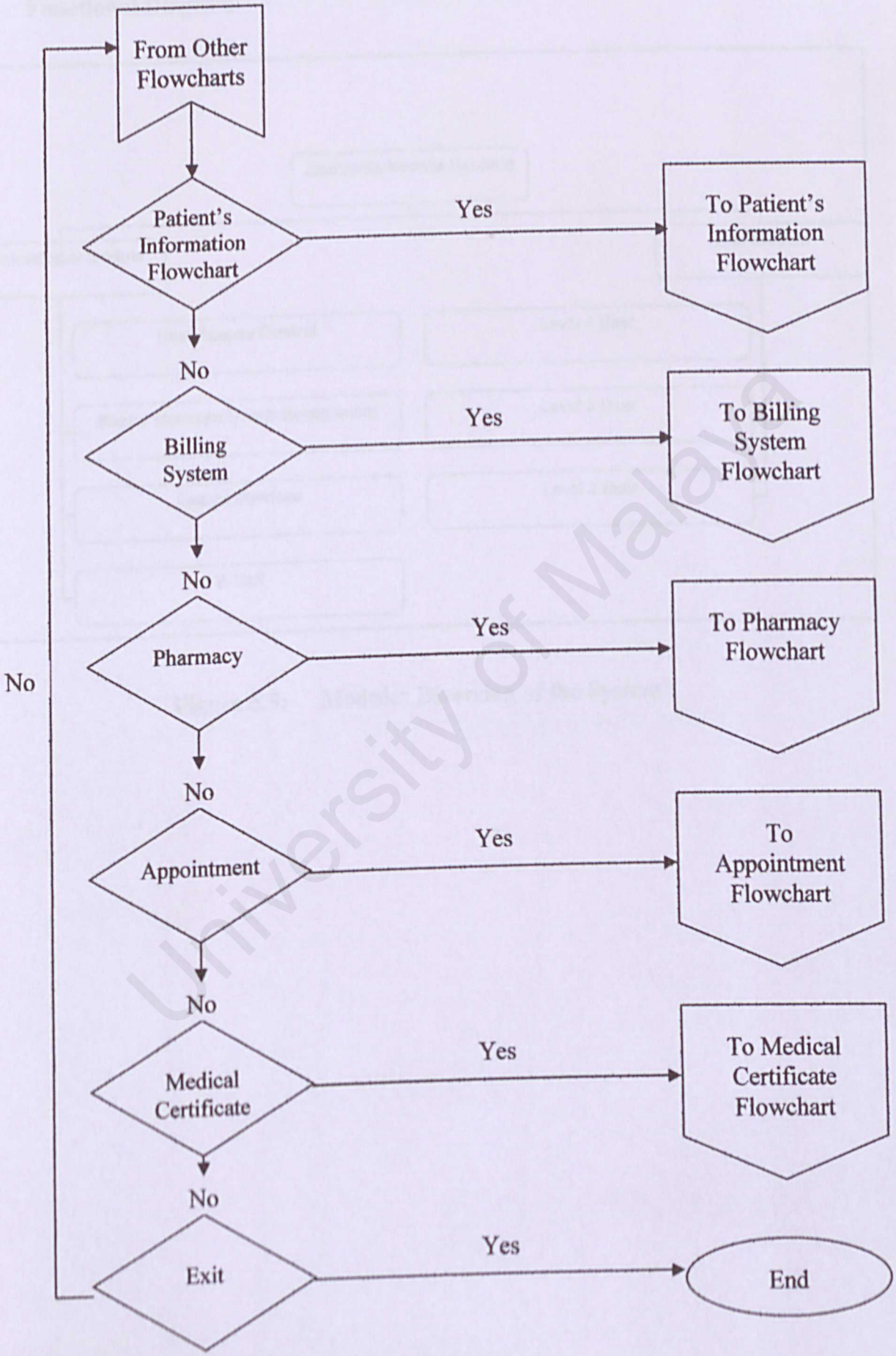


Figure 5.2: Main Screen Flow of Electronic Medical Record

5.2.1 Functional Diagram

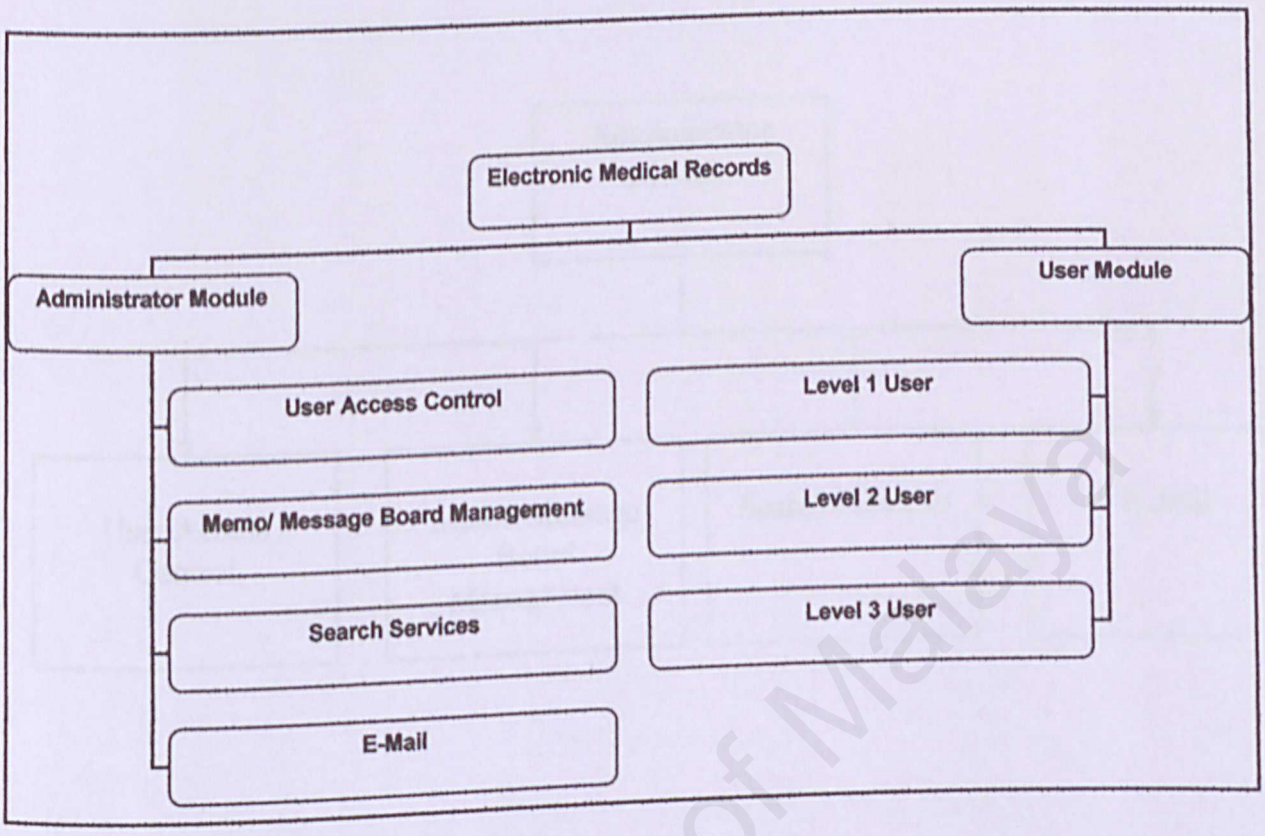


Figure 5.3: Modules Diversion of the System

Functional Diagram For Administrator Module

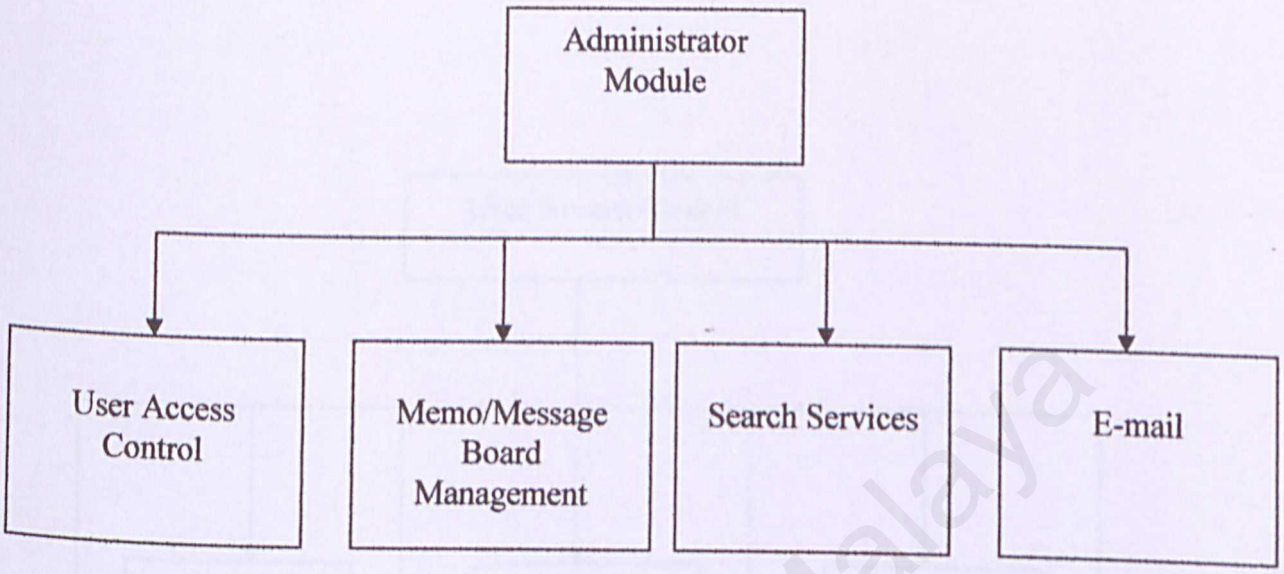


Figure 5.3 : Functional Diagram For Administrator Module

Functional Diagram for User Access Control Sub-process

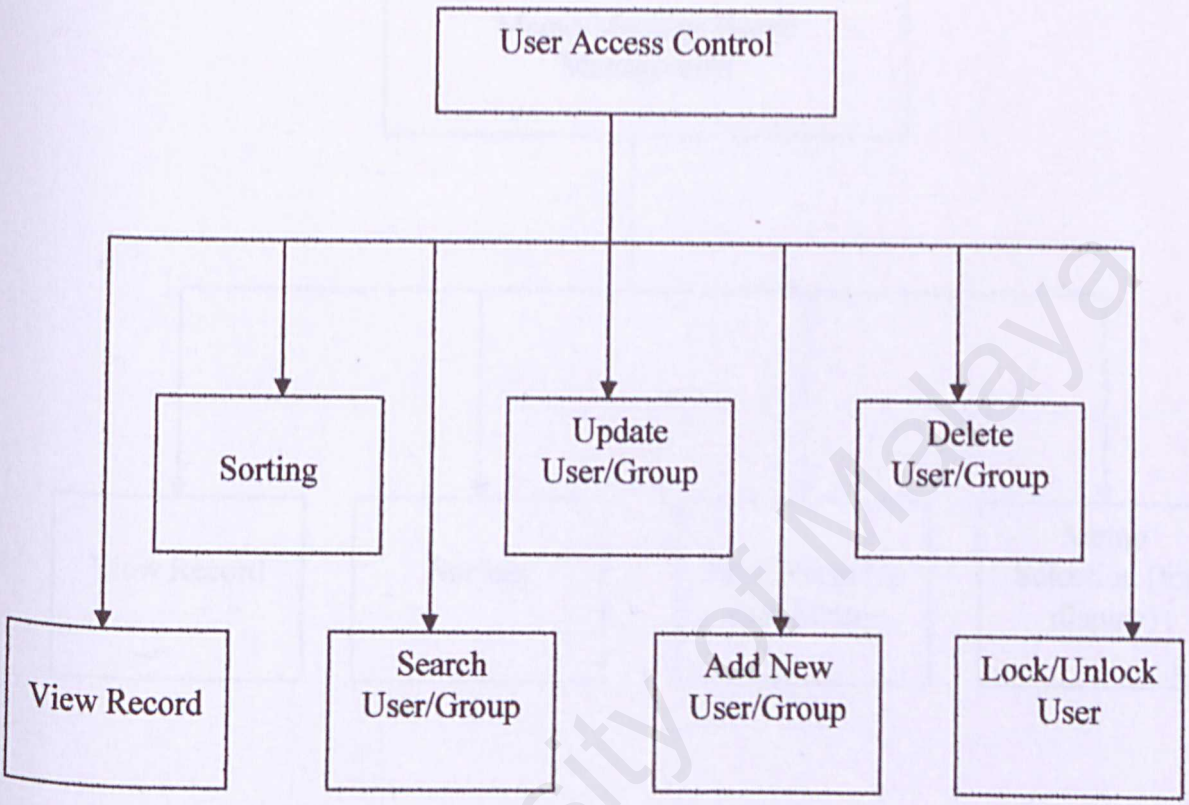


Figure 5.4 : Functional Diagram for User Access Control Sub-process

Functional Diagram For Memo/Message Board Management Sub-process

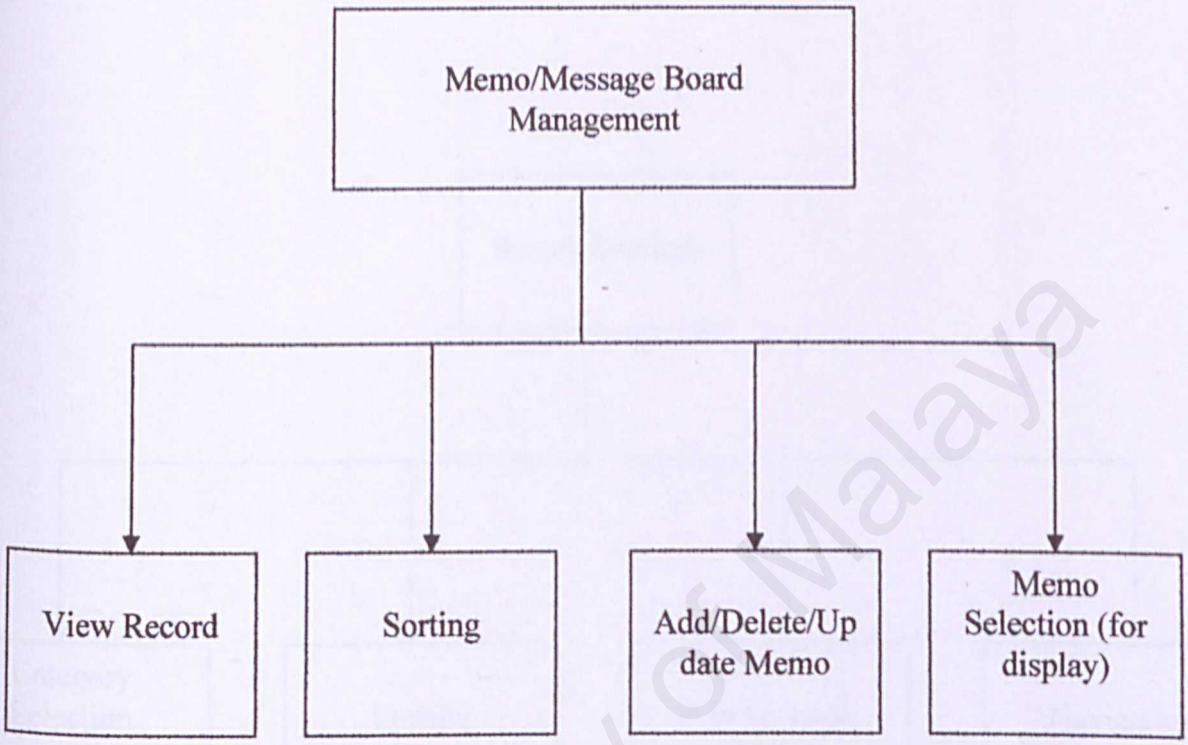


Figure 5.5 : Functional Diagram For Memo/Message Board Management Sub-process

Functional Diagram for Search Services Sub-Processes

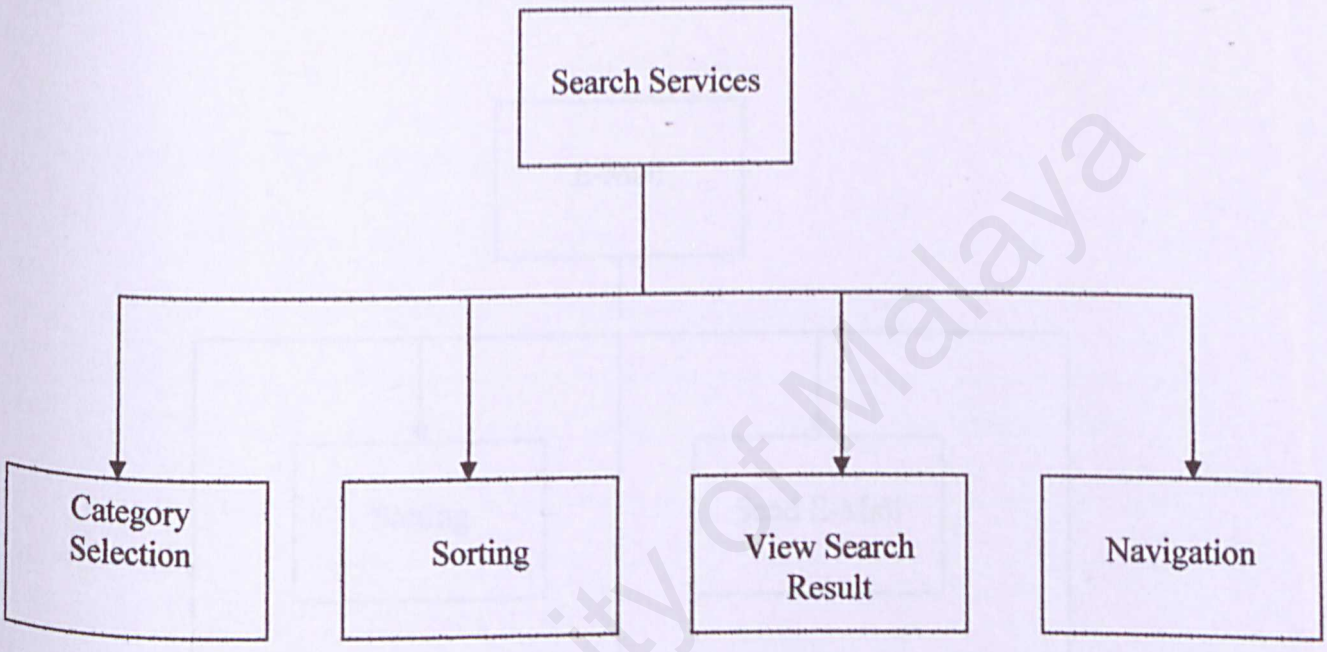


Figure 5.6 : Functional Diagram for Search Services Sub-Processes

Functional Diagram For E-Mail Sub-processes

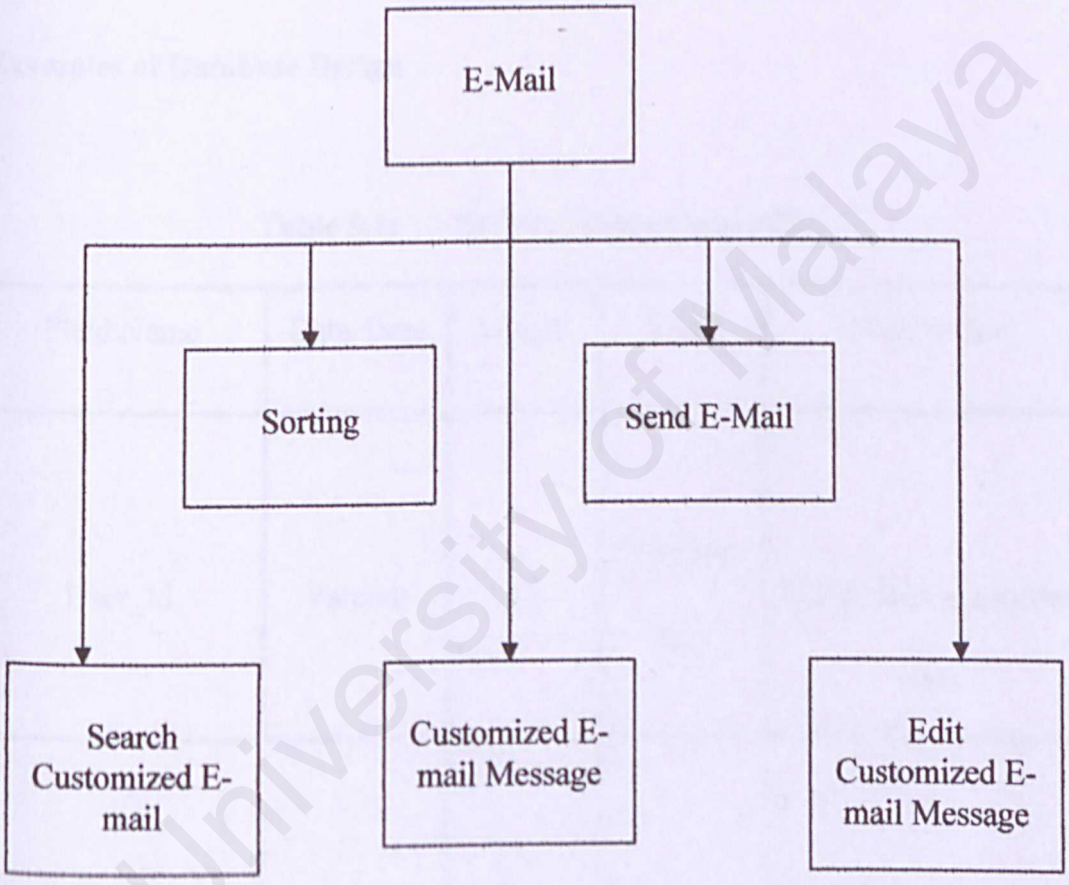


Figure 5.7 : Functional Diagram For E-Mail Sub-processes

5.3 Database Design

Database design is the process of translating logical data models into physical database schemas. The design of the database is very important because it greatly affect the performance of data retrieval, updating and query as well as in the run time period of the system. Besides that a database should be reliable, adaptable and scalable to new unforeseen requirements and applications. The attributes of the database is listed as below:

5.4 : Examples of Database Design

Table 5.1: System Registration table

Field Name	Data Type	Length	Index	Description
User_Id	Varchar	10	Primary Key	The ID for each system user
Login_Password	Varchar	20		System user's login password

Table 5.2: The Patient information table

Field Name	Data Type	Length	Index	Description
Patient_Name	Varchar	150		Patient's Name
Patient_IC	Varchar	20		Patient's IC
Patient_Id	Varchar	20	Primary key	Unique ID that represent each patient
DOB	Date/time	10		Patient's Date of Birth
Age	Int	4		Patient's age
Address	Text	500		Patient's address
OfficePhone	Varchar	20		Patient's Office Phone
Homephone	Varchar	20		Patient's homephone
Mobilephone	Varchar	20		Patient's mobilephone
Postal_code	Varchar	20		Postcode
State	Varchar	50		State
Pager	Varchar	50		Pager
Fax	Varchar	50		Fax
E-mail	Varchar	30		Patient's e-mail
Occupation	Varchar	50		Patient's occupation

Table 5.3: Billing Statement table

Field Name	Data Type	Length	Index	Description
PaymentNo	Varchar	20	Primary key	Unique ID that represent each policy
Balance	Varchar	20		Balance of payment
Patient_Name	Varchar	150		Patient's name
Payer	Varchar	150		Payer's name
Department	Varchar	50		Department's name
Bank_Code	Varchar	25		Bank Code number
Change	Varchar	20		Change of payment
RefNo	Varchar	25		Patient's reference Number
Date	Date/Time	10		Date of payment
Remark	Varchar	500		Patient's remark

Table 5.4: Appointment form table

Field Name	Data Type	Length	Index	Description
Patient_Id	Varchar	10	Primary key	Unique ID that represent each patient
Patient_Name	Varchar	150		Patient's name
Date_appointment	Date/time	10		Date of appointment
Time_appointment	Date/time	10		Time of appointment
Doctor_Id	Varchar	10		Doctor's ID
Doctor_Name	Varchar	150		Doctor's name
Remarks	Varchar	500		Remarks of doctor or patient

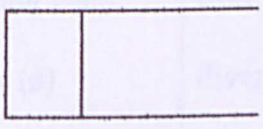
5.5 Data Flow Diagram (DFD) for Electronic Medical Record System

Data Flow Diagram (DFD) will be used to document the process requirements and design of Electronic Medical Records. DFD is a popular process model which depicts the flow of data through a system and the work or processing performed by the system.

The basic notations which are used throughout the DFD documentation are as follows:

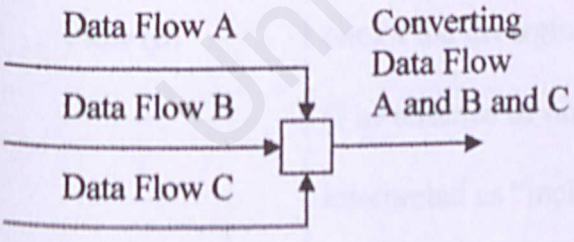
Entity

Process

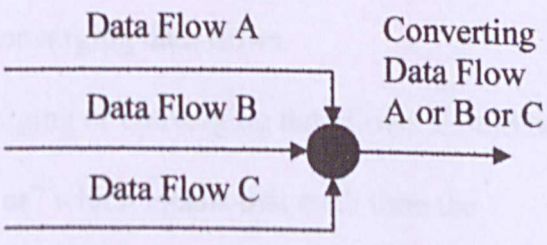


Data Store

Flow of Data



(a)



(b)

Table 5.5 : Description of the basic notations used in DFD

Notations	Description
Entity	An external entity that can send data to and/or receive from the system
Process	Transform the input data to output data
Flow of Data	It represents an input of data to a process, or the output of data from a process. The head of the arrow points toward the data's destination
Data Store	An inventory of data which is implemented as a database
Converging Data Flow (a)	The small square junction means "and". This means that each time the process is performed, it must input (or output) all the diverging or converging data flows
Converging Data Flow (b)	The small black junction means "exclusive or". This means that each time the process is performed; it must input (or output) only one of the diverging or converging data flows. (The absence of one diverging or converging data flows should be interpreted as "inclusive or" which means that each time the process is performed, it may input or output all of the depicted data flows)

5.5.1 Context Diagram for Electronic Medical Records

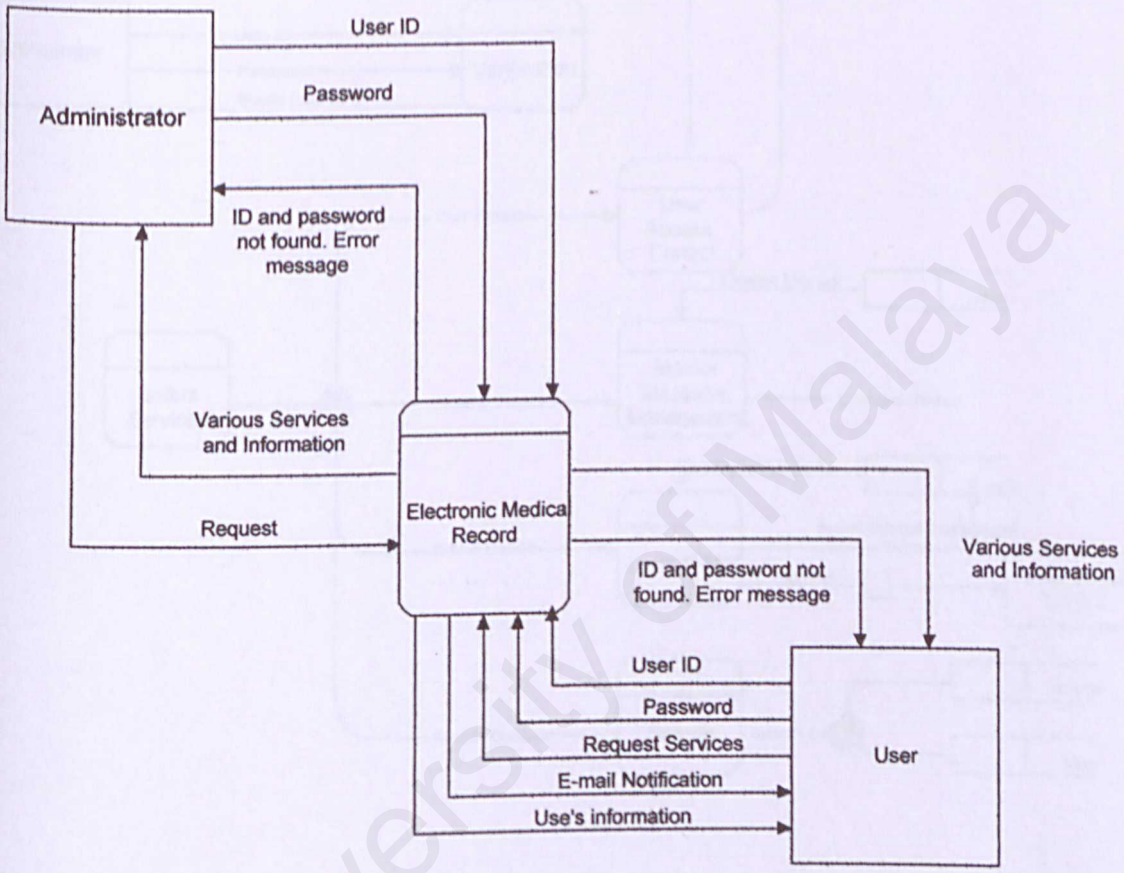


Figure 5.8 : Context Diagram for Electronic Medical Records (Administrator Module and User Module)

5.5.2 Diagram for Administrator Module

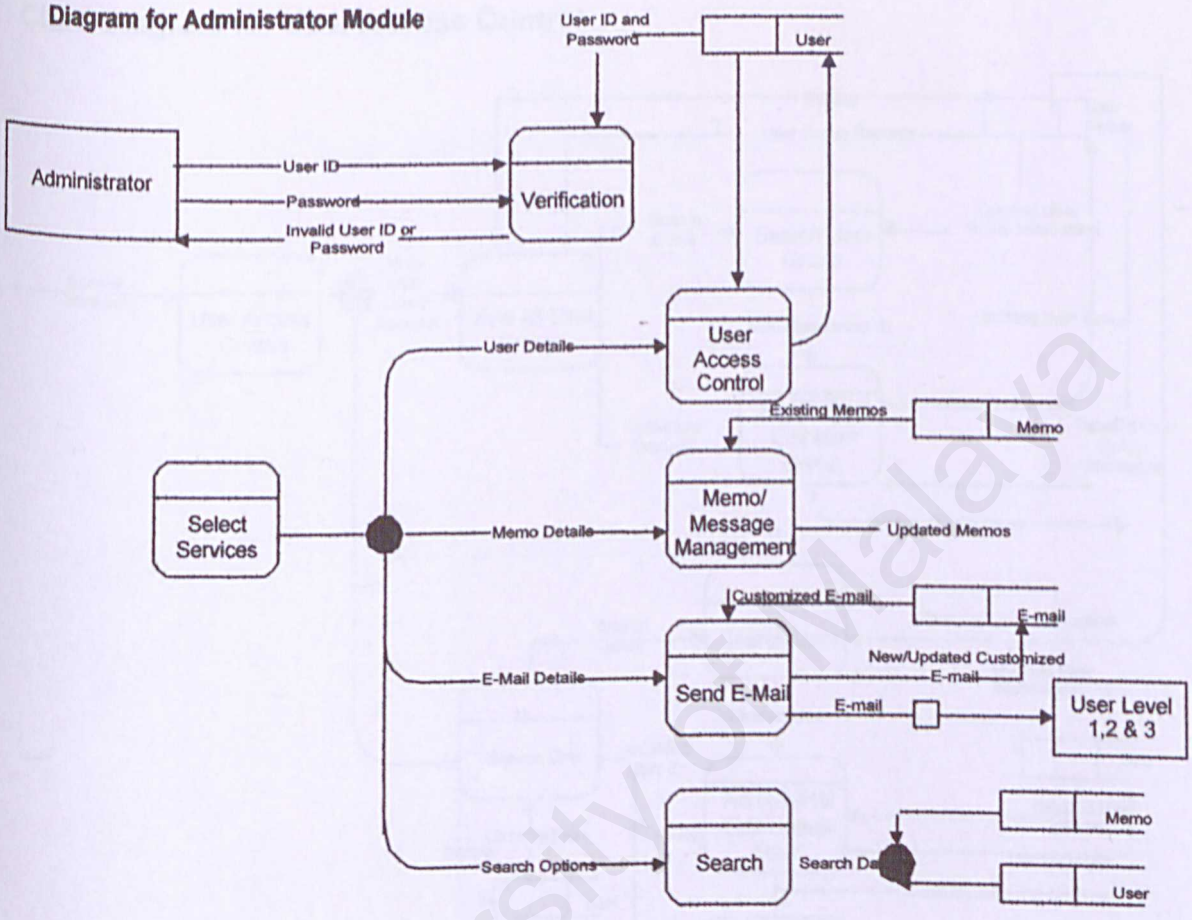


Figure 5.9 : Diagram for Administrator Module

5.5.3 Child Diagram for User Access Control

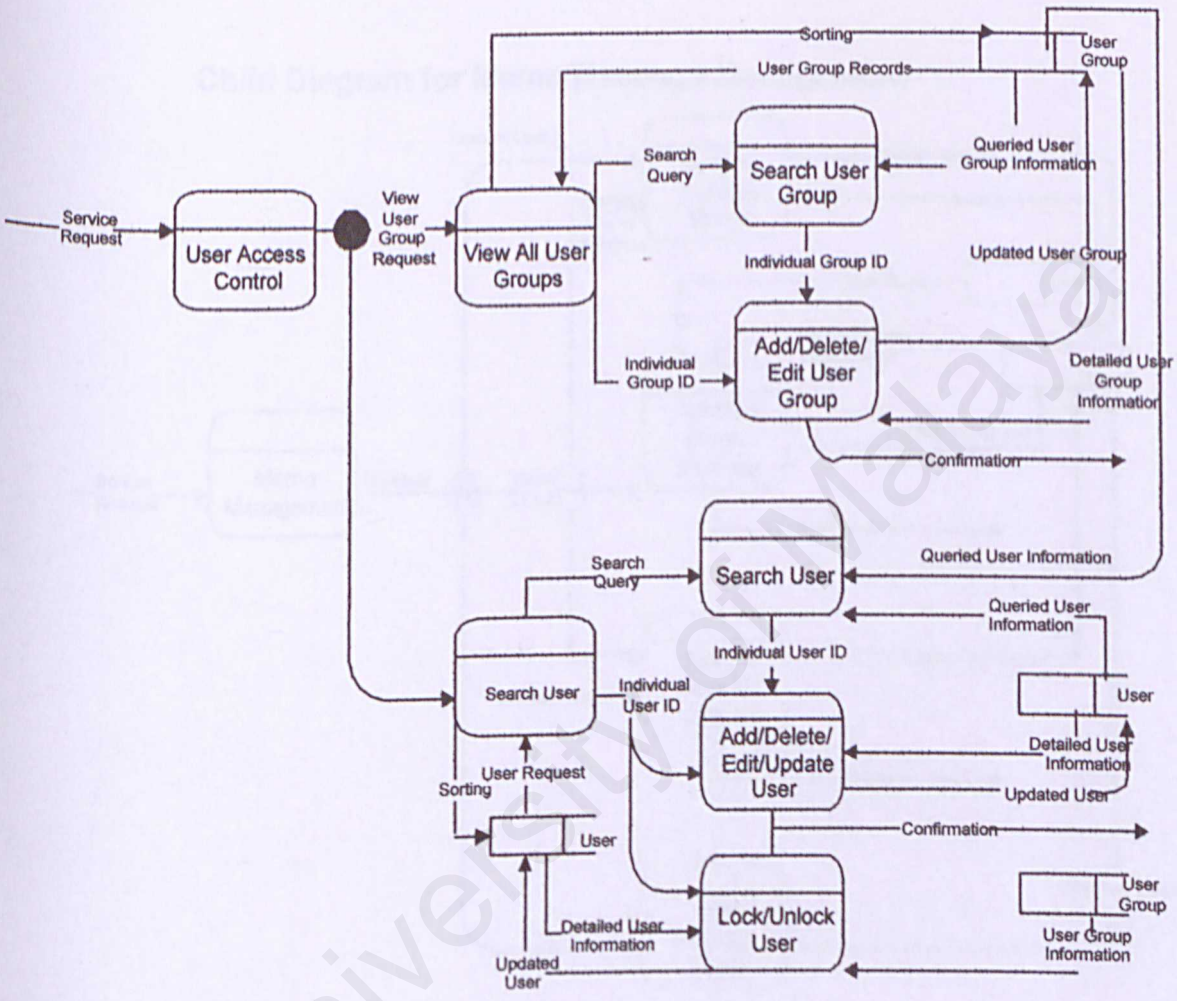


Figure 5.10 : Child Diagram depicting Sub-processes in User Access Control

5.5.4 Child Diagram for Memo/Message Management

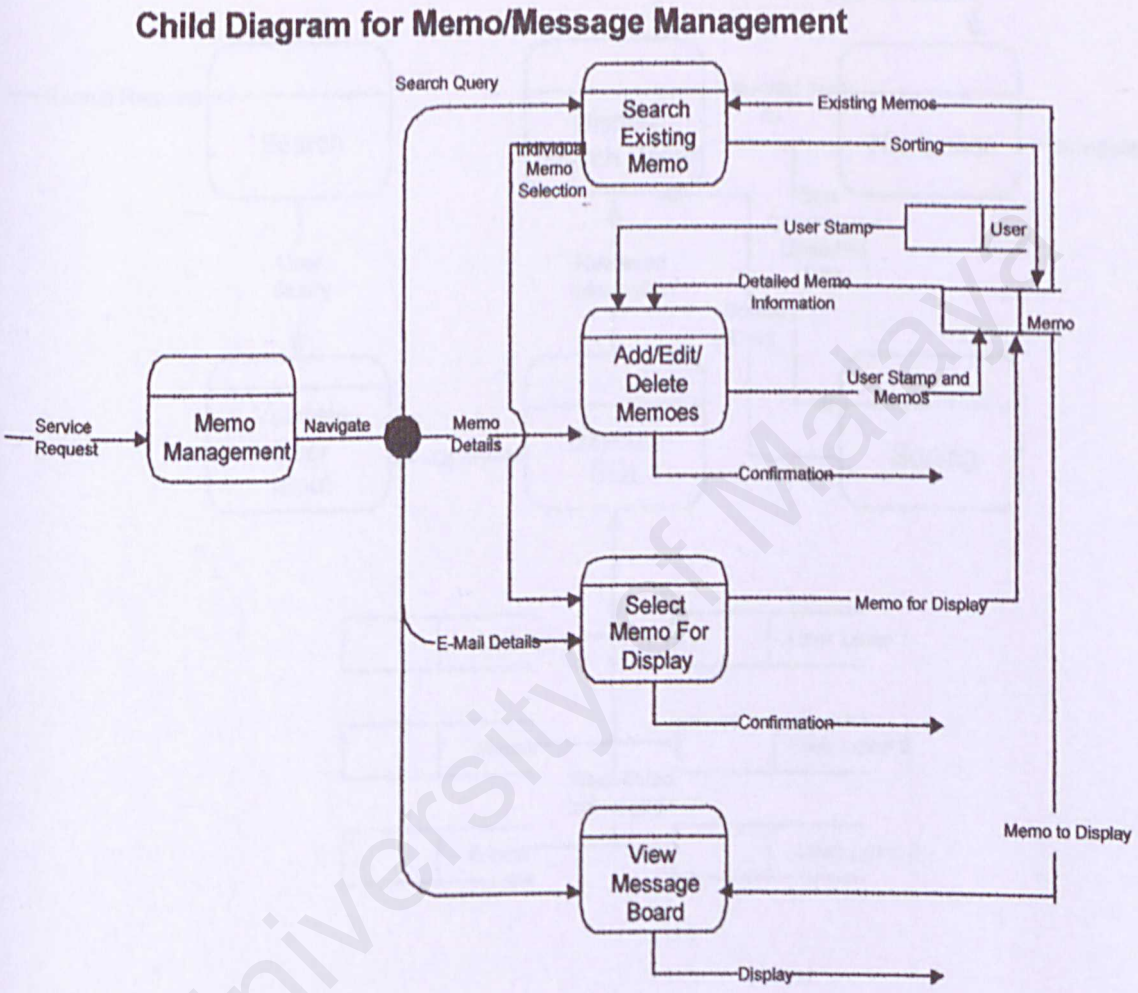


Figure 5.11 : Child Diagram depicting Sub-processes in Memo/Message Board Management

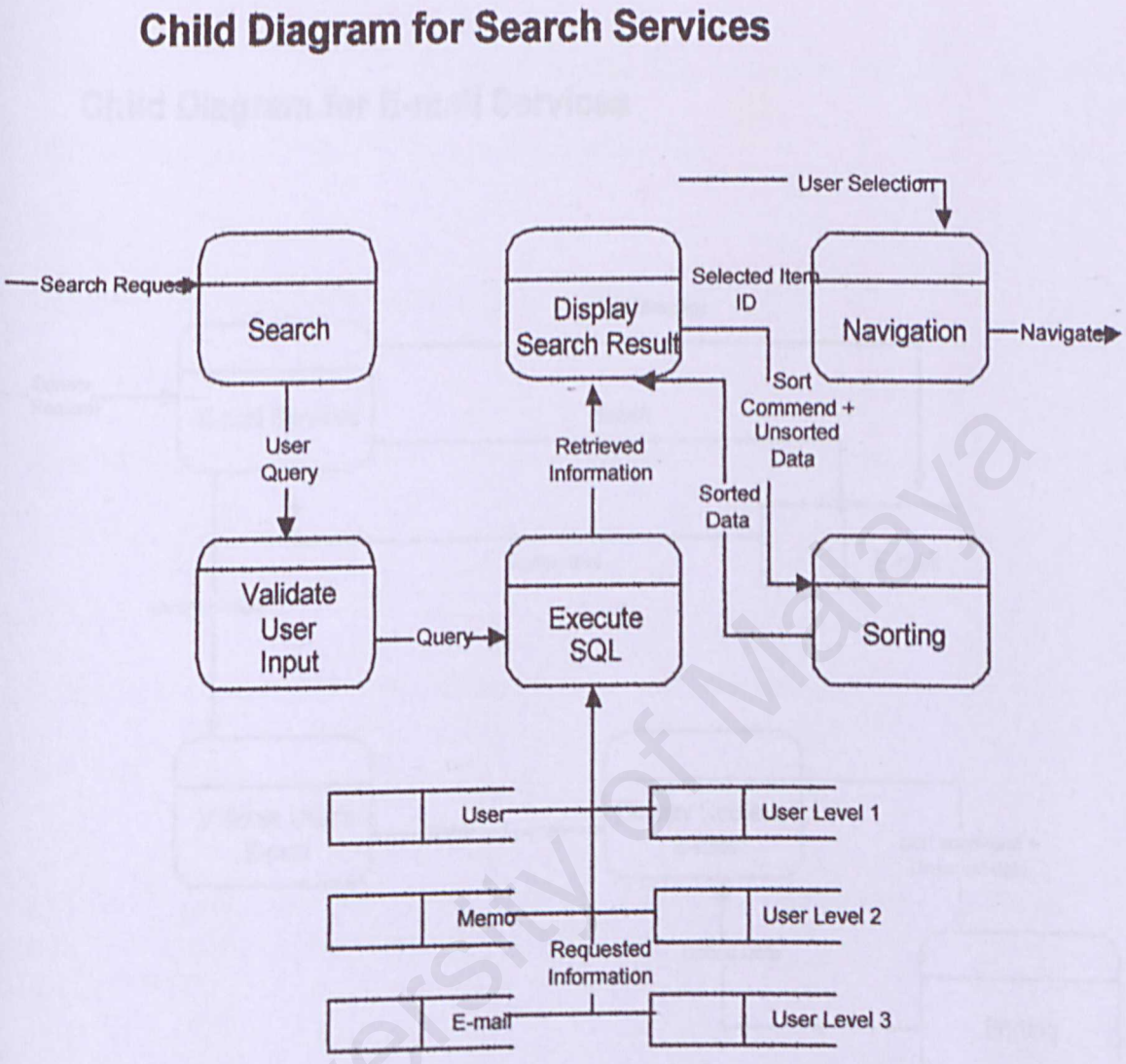


Figure 5.12 : Child Diagram depicting Sub-processes in Search Services

5.5.6 Child Diagram for E-mail

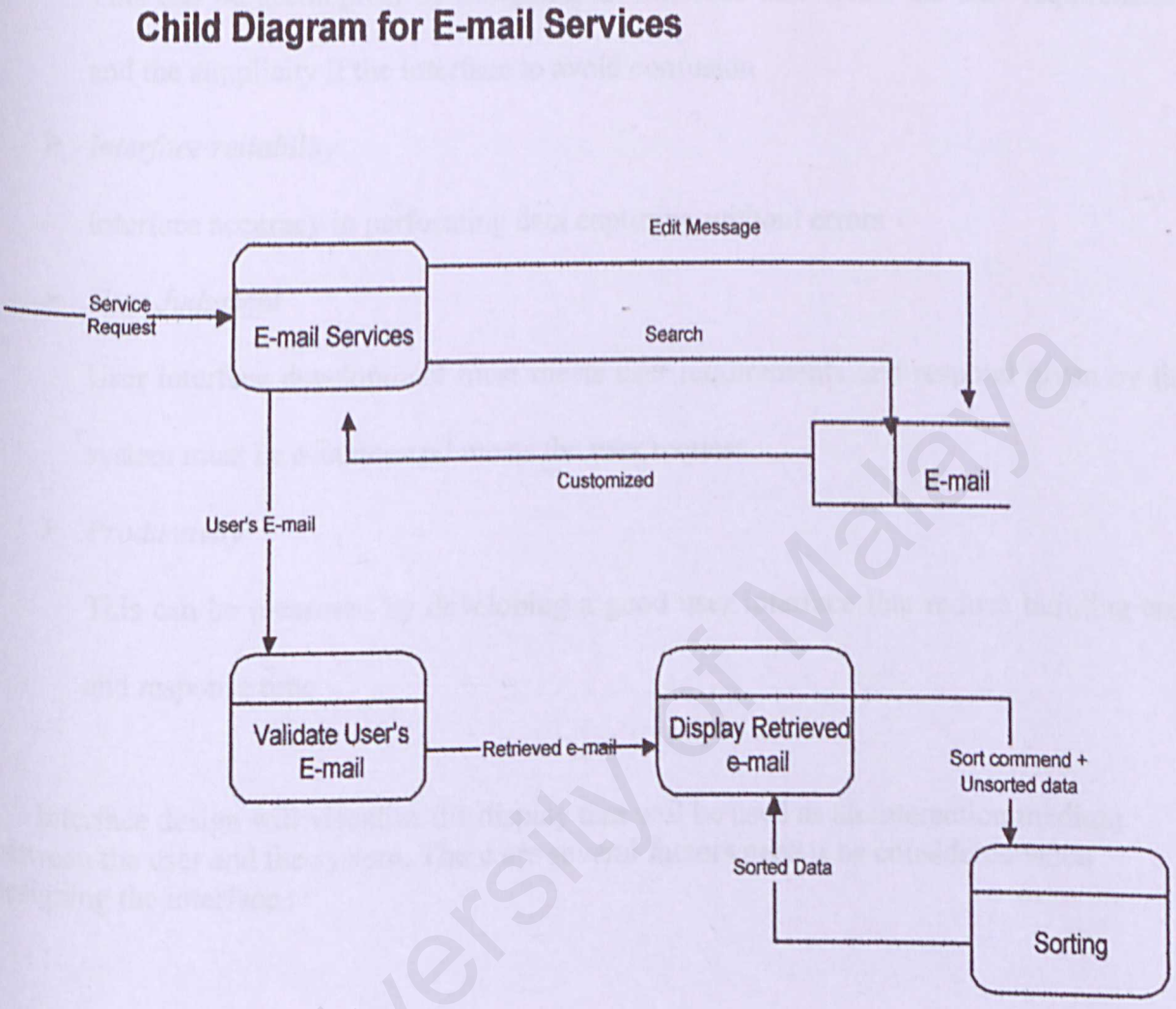


Figure 5.13 : Child Diagram depicting Sub-processes in E-Mail services

5.6 User Interface Design

Objectives of building a user interface:

- *Effectiveness of using user interface*

  This can be accomplish by designing an interface that meets the user requirements and the simplicity if the interface to avoid confusion

- *Interface reliability*

  Interface accuracy in performing data capturing without errors

- *User Judgment*

  User interface development must meets user requirements and respond given by the system must be accurate and meets the user request.

- *Productivity*

  This can be measured by developing a good user interface that reduce building cost and response time.

Interface design will visualize the display that will be used as an interaction medium between the user and the system. There are several factors need o be considered when designing the interface :

1. Soft background color, icons, logos, pictures and appropriate fonts.
2. Consistency between pages to avoid confusion for users.
3. Control measures such as combo box selection box and check box are used to accelerate data recording process.

4. Suitable types of fonts are used so the interface would be more organized and easy to understand.

5.6.1 Examples of User Interface Design

5.6.1.1 Example of user interface design for registration login of the system

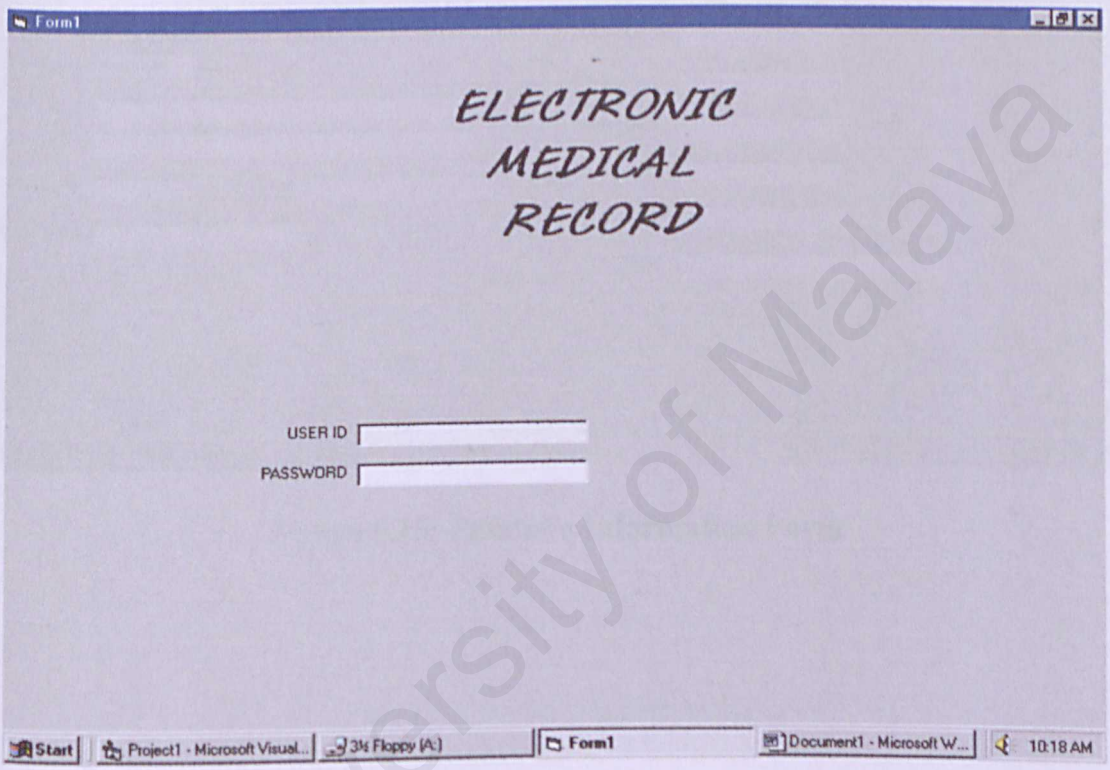


Figure 5.14: Registration Page

5.6.1.2 Example of User interface design for patient registration information

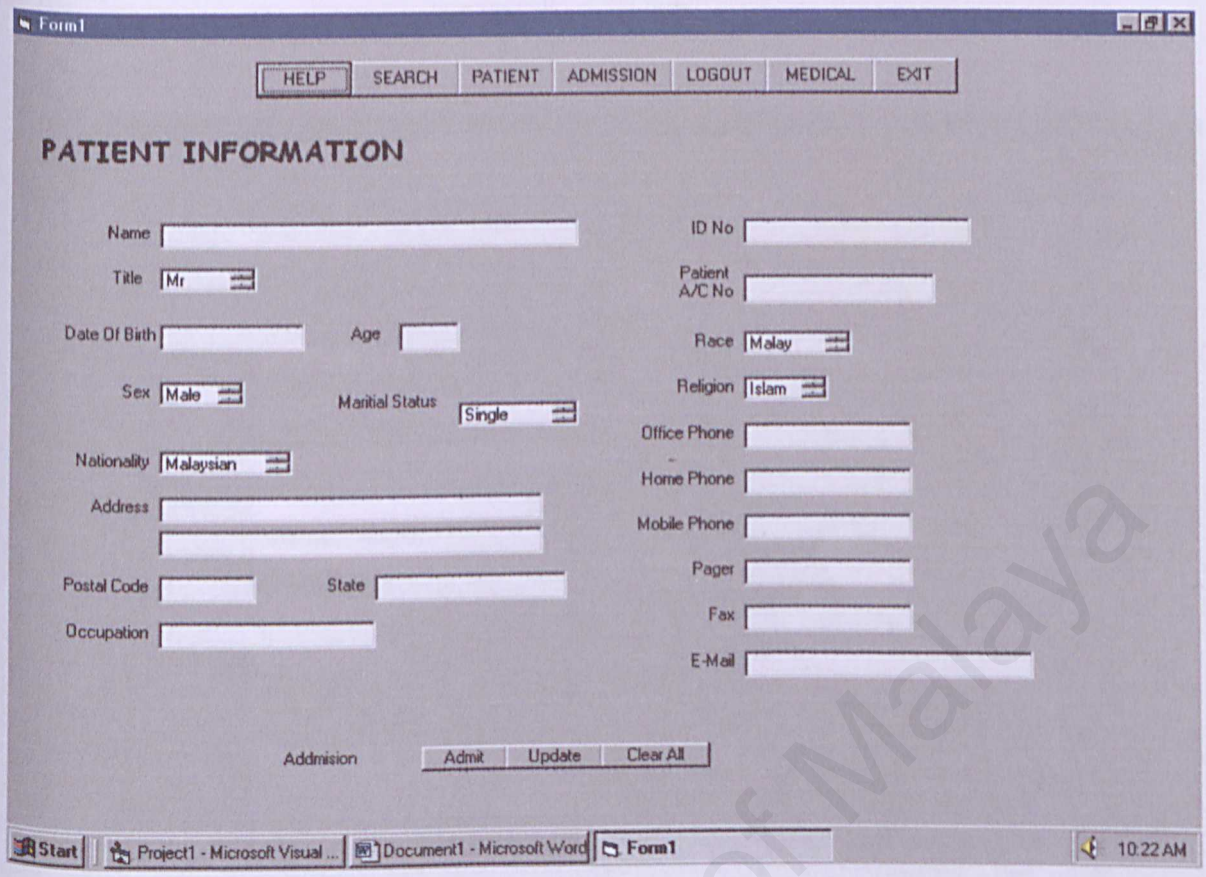


Figure 5.15: Patient's Information Form

5.6.1.3 Example of user interface design for Billing Statement

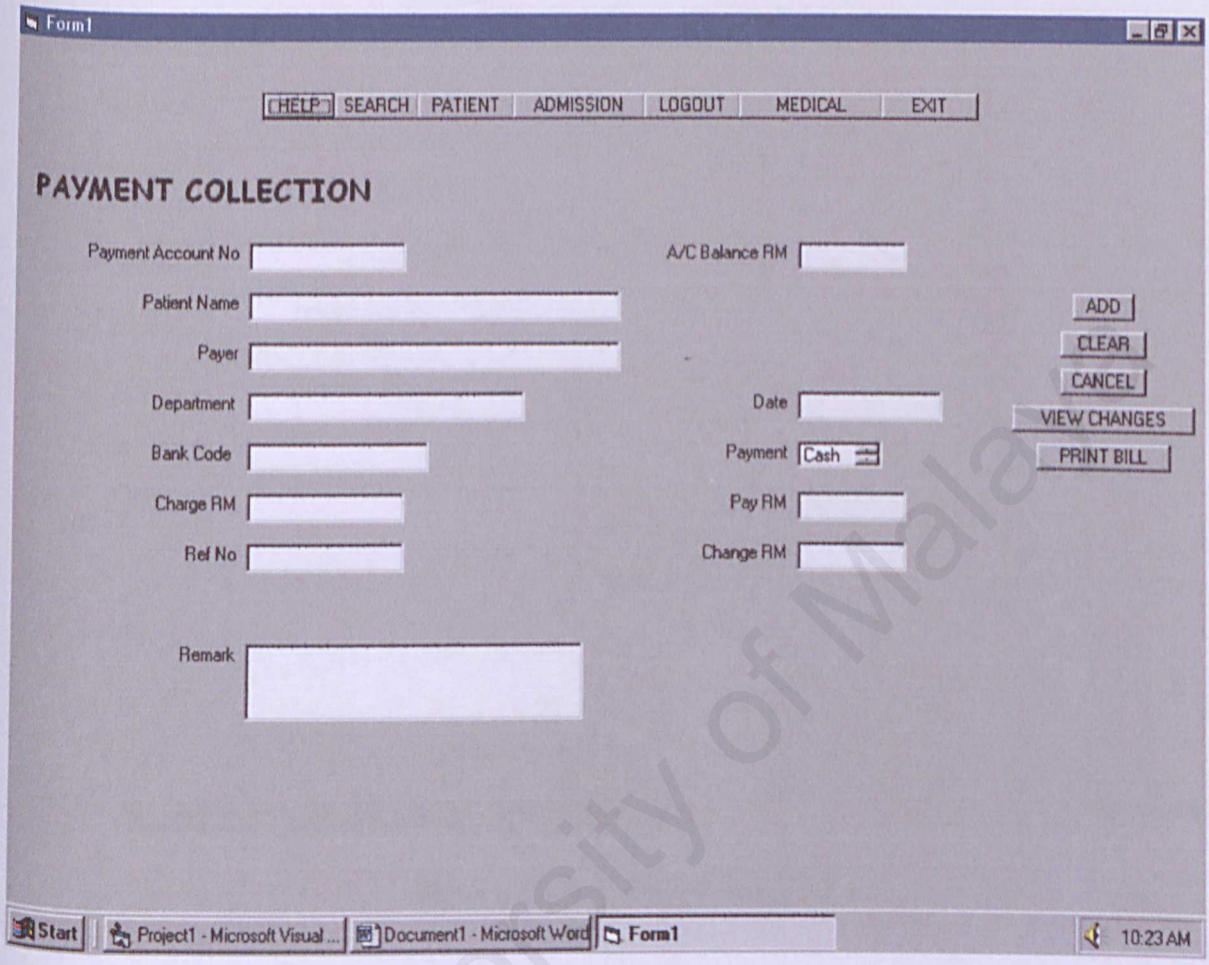


Figure 5.16: Billing Statement

5.6.1.4 Example of User interface design for Appointment form

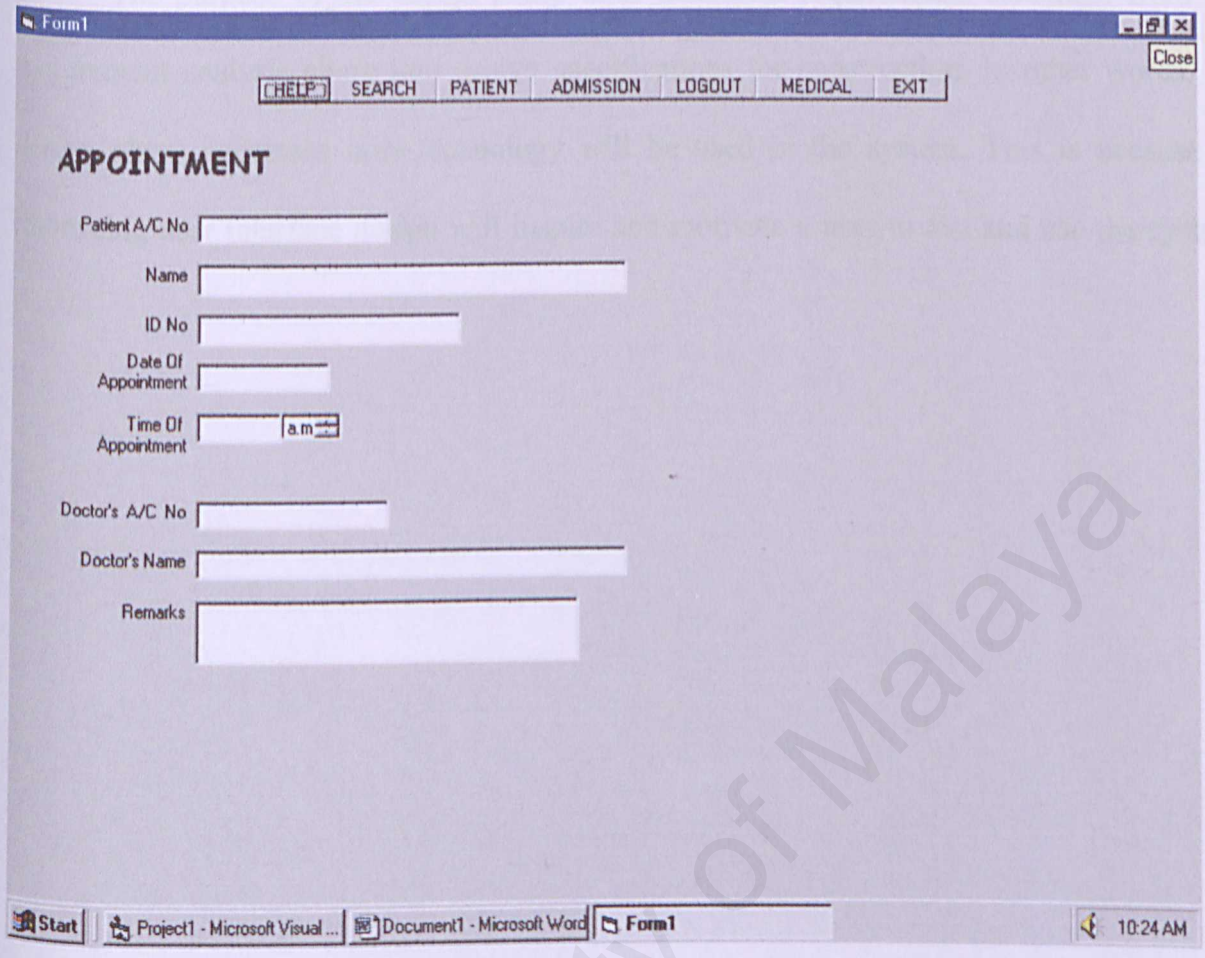


Figure 5.17: Appointment Form

5.7 Summary

The purpose of the design phase is to transform requirements statement from the requirement analysis phase into design specifications for construction. In other words, the design phase addresses how technology will be used in the system. This is because an interesting user interface design will inspire and motivate a user to test and use the system.

CHAPTER 6 SYSTEM IMPLEMENTATION

6.0 Introduction

After the system designing phase on how the system should be functioning, the next process will involves the implementation phase. System implementation is a process that converts the system requirements and designs into program codes. In a software project, the requirements analysis, system design and implementation phases do not have a clear boundary. Each phase tends to overlap one another. This phase at times involves some modifications to the previous design. The implementation phase is an important element especially when it involves a project developed by a team of people where integration of system, works by different people takes a huge effort.

The design phase earlier in the system life cycle is directed towards a final objective which is to translate the concept of the system into a software representation that is understood by the computer. The coding process involves transforming of the design into a programming language. The effort spent in this phase will actually determines the success of the system and ease the processes of modification, debugging, testing, verification, system integration and for future enhancement.

6.1 Development Environment

Development environment plays a major role in determining the speed of developing the system. Using the suitable hardware and software will not only help to speed up the system development but also determine the success of the project. After implementing the system, the requirement of hardware and software that was stated in the previous two chapter (Chapter 3 System Analysis) can be finalized. The final list of the hardware and software tools used to develop the entire XML browser based electronic medical record is listed below.

Adobe Photoshop 6.0	User Interface Design	Image design and creation
Microsoft Photo Editor	User Interface Design	Image design and creation
Microsoft Word 2000	Documentation	Design and writing Report Documentation

Table 6.1 Software Tools Requirements

6.2 Program Development and Coding

Program development is the process of creating the programs needed to satisfy an information system's processing requirements. Developing and Coding is the phase which takes the longest time in the development life cycle. Therefore, using the right tool and the right way to develop the system are crucial in determining the success of a project. For Electronic Medical Record, it involves developing using different programming languages from Active Server Pages (ASP), Macromedia Dreamweaver MX, Adobe Photoshop 7.0 and Hypertext Markup Language (HTML). Before starting on the coding process or any other detailed works on the program, a review on the program documentation needs to be done followed by design of the program and finally going into the program coding process.

6.2.1 Review the Program Documentation

The first and foremost step to be taken in program development phase is to review the program documentation that was prepared during the earlier phases. The program documentation prepared in the system design phase of Electronic Medical Record consists of architectural view, concepts and controls, module flow diagram and also the sample layout of the interface. The documentation provides a guide and an understanding of the works that need to be done in the coding phase

6.2.2 Designing the Program

After reviewing the program documentation, designing the program is the next following process after that. For this phase, determining how the program can accomplish the features and functions that are described in the program documentation and developing a logical solution to the programming problem is done. The logical solution or the logic of the program is a step-by-step solution to the programming problems.

6.2.3 Coding Approaches

There are two approaches in coding, namely top-down and bottom-up. The bottom-up coding is based on coding some complete lower level modules and leaving the high-level modules merely as skeletons that are used to call the lower modules, whereas the top-down approach is the reverse.

XML browser based electronic medical record system was developed modularly using both the top-down and bottom-up approaches. Developing Electronic medical record with top-down approach involves building the high-level software modules that are refined into functions and procedures. The advantages of using bottom-up approach in Electronic Medical Record System are:

1. Testing can begin on some of the modules while others are still being coded.
2. Critical functions can be coded first to test their efficiency.

6.2.4 Coding Style

Coding style is an important attribute of source code. An easy to read source code makes the system easier to maintain and enhance. Elements taken into considerations while coding an easy to maintain and enhance system are internal documentation, standard naming convention and standard graphical user interface.

Internal documentation is achieved by using comments while coding, providing a clear guide to programmers for future enhancement. Statements of purpose indicating the functions of modules and descriptive comment are embedded into source code to describe the processing functions.

A standard naming convention and also a standard usage of graphical user interface components is employed in developing the system making. Standard naming convention provides programmers with easy identification of variables. While a standard in usage of graphical user interface components provides the users an environment that will not generate much surprise to them. Usages of these standards perform as a mean towards coding consistency and standardisation.

An Active Server Page is primarily a scripting environment. Languages used to develop an ASP are HTML and Macromedia Dreamweaver MX. The challenge of coding in ASP is of determining and separating the HTML source code from the scripting counterpart.

6.2.5 Hyper Text Markup Language (HTML)

Using a markup language designed for simplicity and with a flexible structure, HTML allowed text and graphics to be displayed in a Web browser or other suitably enabled application. All the HTML coding normally are delimited by the <BODY>……</BODY>tags. Certain parts of the document could become hyperlinks, which – when click – caused a different page or a different section of the same page to be displayed.

6.2.6 Macromedia Dreamweaver MX

Macromedia Dreamweaver MX is a professional HTML editor for designing, coding, and developing websites, web pages, and web applications. Whether enjoy the control of hand-coding HTML or prefer to work in a visual editing environment, Dreamweaver provides with helpful tools to enhance web creation experience.

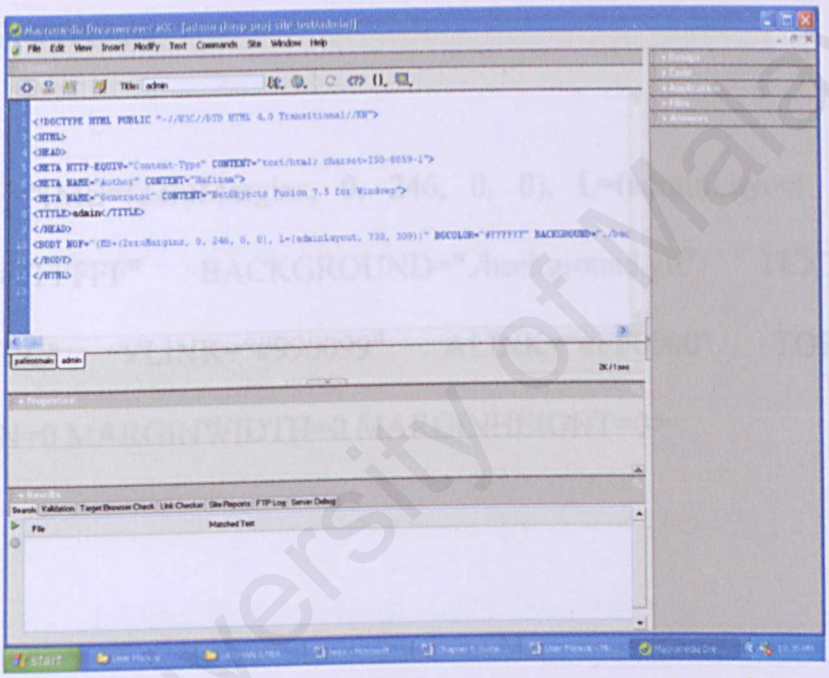


Figure 6.1 Work space for Macromedia Dreamweaver MX

The visual editing features in Dreamweaver letting quickly create pages without writing a line of code. Dreamweaver also helps to build dynamic database-backed web applications using server languages such as ASP, ASP.NET, ColdFusion Markup Language (CFML), JSP, and PHP.

Below is an example taken from the project with file name as "admin.asp"

```
!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=ISO-8859-1">
<META NAME="Author" CONTENT="Hafizam">
<META NAME="Generator" CONTENT="NetObjects Fusion 7.5 for Windows">
<TITLE>admin</TITLE>
</HEAD>
<BODY NOF="(MB=(ZeroMargins, 0, 246, 0, 0), L=(adminLayout, 730, 309))"
BGCOLOR="#FFFFFF" BACKGROUND="./background.gif" TEXT="#000000"
LINK="#0033CC" VLINK="#990099" ALINK="#FF0000" TOPMARGIN=0
LEFTMARGIN=0 MARGINWIDTH=0 MARGINHEIGHT=0>
</BODY>
</HTML>
```

Admin Change Password

Below is an example taken from the project with file name as “adminpasschange.asp”:

```
If Session("adminLoggedIn") <> "true" Then
  Response.Redirect("adminlogin.asp")
End If
%>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=ISO-8859-1">
<META NAME="Author" CONTENT="Hafizam">
<META NAME="Generator" CONTENT="NetObjects Fusion 7.5 for Windows">
<TITLE>adminpasschange</TITLE>
<NOLAYER>
<STYLE ID="NOF_STYLE_SHEET">
<!--
```

New Patient

Below is an example taken from the project with file name as "new patient.asp":

```
'**********generating Patient ID*****************
set con=Server.CreateObject("ADODB.Connection")
    sConnString = "DRIVER={Microsoft Access Driver (*.mdb)};" & _
                  "DBQ=" & Server.MapPath("dbHospital.mdb") & ";"
                    con.Open(sConnString)

                    set maxPatRs=Server.CreateObject("ADODB.Recordset")
                    maxPatRs.open "Select max(idsufix) from tblPatients",con,1,3
                    if isnull(maxpatrs(0)) then
                      varSufix=1
                    else
                      varSufix=maxpatrs(0)+1
```

```
                end if
                varPrefix="P" & right(year(date),2)
                varNewPatId= varPrefix & varSufix
**********END Generating Patient ID**********
```

View Patient Database in XML browser-based

Below is an example taken from the project with file name as "patdb.asp":

```
'Name of the access db being queried
accessdb="dbHospital.mdb"

'Connection string to the access db
cn="DRIVER={Microsoft Access Driver (*.mdb)};"
cn=cn & "DBQ=" & server.mappath(accessdb)

'Create a server recordset object
Set rs = Server.CreateObject("ADODB.Recordset")


'Query the states table from the state_info db
sql = "select * from tblPatients "


'Execute the sql
rs.Open sql, cn
```

```
' Move to the first record

rs.MoveFirst

' Name for the ouput document

file_being_created= "patients.xml"

' create a file system object

set fso = createobject("scripting.filesystemobject")

' create the text file - true will overwrite any previous files

' Writes the db output to a .xml file in the same directory

Set act = fso.CreateTextFile(server.mappath(file_being_created), true)

' All non repetitive xml on top goes here

act.WriteLine("<?xml version=""1.0""?>")

act.WriteLine("<Patients>")

'Loop to output all the query results to the xml document

do while not rs.eof

' counter to give each record a sequential listing

counter=counter+1
```

```
act.WriteLine("<Patient id="""& rs("patid") &""">")

'act.WriteLine("<Patient_id>" & rs("patid") & "</Patient_id>" )

act.WriteLine("<Patient_name>" & rs("patName") & "</Patient_name>" )

act.WriteLine("<Address>" & rs("address") & "</Address>")

act.WriteLine("<Phone>"& rs("phone") & "</Phone>")

act.WriteLine("<E-mail>"& rs("email") & "</E-mail>")

act.WriteLine("</Patient>")


' move to the next record

rs.movenext

loop


' All non repetitive xml on bottom goes here

act.WriteLine("</Patients>")



' close the object (xml)

act.close


' Writes a link to the newly created xml document in the browser
```

```
response.write "<a href='patients.xml'>Click here to see Patients</a> (.xml) Database
<br>"
response.write "on " & now() & "<br>"
%>
```

Modify / Delete a Patient

Below is an example taken from the project with file name as "viewpats.asp":

```
response.write "<center><br>"
set con=Server.CreateObject("ADODB.Connection")
    sConnString = "DRIVER={Microsoft Access Driver (*.mdb)};" & _
                  "DBQ=" & Server.MapPath("dbHospital.mdb") & ";"
                con.Open(sConnString)
set viewPatRs=Server.CreateObject("ADODB.Recordset")
                viewPatRs.open "Select * from tblPatients",con,1,3
                if viewPatRs.recordcount<1 then
                    response.write "<font color=red><b>No Patients found</b>"
                else
                    response.write "<table border=1 cellspacing=0
bordercolor=black><tr><Th>Patient ID<th>Patient
Name<th>Address<th>Phone<Th>E-mail<th>Action"
                do while not viewPatRs.eof
                    response.write "<tr><td>" & viewPatRs("patId")
```

```
                    response.write "<td>" & viewPatRs("patName")
                    response.write "<td>" & viewPatRs("address")
                    response.write "<td>" & viewPatRs("phone")
                    response.write "<td>" & viewPatRs("email")
                    response.write "<Td> <a href=modipats.asp?patid=" &
Server.URLEncode(viewPatRs("patId")) & ">Modify</a> / <a href=delpat.asp?patid="
& Server.URLEncode(viewPatRs("patId")) & ">Delete</a>"
                    viewpatrs.movenext
                   loop
                end if
response.write "</table>"
%>
```

New Bill

Below is an example taken from the project with file name as "billing.asp":

```
set con=Server.CreateObject("ADODB.Connection")
    sConnString = "DRIVER={Microsoft Access Driver (*.mdb)};" & _
                  "DBQ=" & Server.MapPath("dbHospital.mdb") & ";"
                con.Open(sConnString)
set patrs=Server.CreateObject("ADODB.Recordset")
                patrs.open "Select * from tblPatients",con,1,3
                if patrs.recordcount<0 then
```

```
                response.write "<font color=red><b>Sorry, currently, No Patinet is
available</b>"
            else
                response.write "<select name=cmbPat>"
                do while not patrs.eof
                    response.write "<option value=" & patrs("patid") & ">" &
patrs("patid") & "</option>"
                patrs.movenext
                loop
            end if
```

View Billing

Below is an example taken from the project with file name as "viewbill.asp":

```
            set viewBillRs=Server.CreateObject("ADODB.Recordset")
            viewBillRs.open "Select * from tblBill",con,1,3
            if viewBillRs.recordcount<1 then
                response.write "<font color=red><b>No Bills found</b>"
            else
                response.write "<table border=1 cellspacing=0
bordercolor=black><tr><Th>Bill ID<th>Patinet
ID<th>Amount<th>Date<Th>Remarks<th>Action"
            do while not viewBillRs.eof
                response.write "<tr><td>" & viewBillRs("BillId")
```

```
                response.write "<td>" & viewBillRs("patid")
                response.write "<td>" & viewBillRs("amount")
                response.write "<td>" & viewBillRs("bdate")
                response.write "<td>" & viewBillRs("remarks")
                response.write "<Td> <a href=delBill.asp?Billid=" &
Server.URLEncode(viewBillRs("BillId")) & ">Delete</a>"
                viewBillrs.movenext
            loop
        end if
response.write "</table>"
```

New Certificate

Below is an example taken from the project with file name as "newcerti.asp":

```
set patrs=Server.CreateObject("ADODB.Recordset")
        patrs.open "Select * from tblPatients",con,1,3
        if patrs.recordcount<0 then
            response.write "<font color=red><b>Sorry, currently, No Patinet is
available</b>"
        else
            response.write "<select name=cmbPat>"
            do while not patrs.eof
                response.write "<option value=" & patrs("patid") & ">" &
patrs("patid") & "</option>"
```

```
                patrs.movenext
            loop
            end if
```

View / Delete a Certificate

Below is an example taken from the project with file name as "viewcert.asp":

```
set viewCertRs=Server.CreateObject("ADODB.Recordset")
            viewCertRs.open "Select * from tblcert",con,1,3
            if viewCertRs.recordcount<1 then
                response.write "<font color=red><b>No Certificates found</b>"
            else
            response.write "<table border=1 cellspacing=0
bordercolor=black><tr><Th>Certificate ID<th>Patinet
ID<th>Date<Th>Remarks<th>Action"
            do while not viewCertRs.eof
                response.write "<tr><td>" & viewCertRs("CertId")
                response.write "<td>" & viewCertRs("patid")
                response.write "<td>" & viewCertRs("cdate")
                response.write "<td>" & viewCertRs("remarks")
                response.write "<Td> <a href=delCert.asp?Certid=" &
Server.URLEncode(viewCertRs("CertId")) & ">Delete</a>"
                viewCertrs.movenext
            loop
```

```
            end if
response.write "</table>"
```

New Doctor

Below is an example taken from the project with file name as "newdoc.asp":

```
set maxDocRs=Server.CreateObject("ADODB.Recordset")
                maxDocRs.open "Select max(idsufix) from tblDoc",con,1,3
                if isnull(maxDocrs(0)) then
                  varSufix=1
                else
                  varSufix=maxDocrs(0)+1
                end if
                varPrefix="D" & right(year(date),2)
                varNewDocId= varPrefix & varSufix
```

View / Modify / Delete a doctor

Below is an example taken from the project with file name as "viewdocs.asp":

```
set viewDocRs=Server.CreateObject("ADODB.Recordset")
                viewDocRs.open "Select * from tblDoc",con,1,3
                if viewDocRs.recordcount<1 then
```

```
                response.write "<font color=red><b>No Doctor found</b>"
            else
            response.write "<table border=1 cellspacing=0
bordercolor=black><tr><Th>Doctor ID<th>Doctor
Name<th>Address<th>Phone<Th>E-mail<th>Action"
            do while not viewDocRs.eof
                response.write "<tr><td>" & viewDocRs("DocId")
                response.write "<td>" & viewDocRs("DocName")
                response.write "<td>" & viewDocRs("address")
                response.write "<td>" & viewDocRs("phone")
                response.write "<td>" & viewDocRs("email")
                response.write "<Td> <a href=modiDocs.asp?Docid=" &
Server.URLEncode(viewDocRs("DocId")) & ">Modify</a> / <a
href=delDoc.asp?Docid=" & Server.URLEncode(viewDocRs("DocId")) &
">Delete</a>"
             viewDocrs.movenext
                loop
            end if  response.write "</table>"
```

6.2.7 Summary

To sum up chapter 6, we can say that the implementation of Electronic Medical Record system focus on various aspects which governs the functionality of the system to fulfill the functional and non functional requirements of the system. Hence the choosing of development platform which ranges from the operating system till the programming platform chosen has to be up most suitable to the requirement of the system. The coding approach meanwhile focus on giving the best reliability and performance of the system and by the end of the day the Electronic Medical Record system can be say to be developed by the most suitable development platform with the most robust and flexible coding design focus on reliability and performance

CHAPTER 7 SYSTEM TESTING

7.0 Introduction

System testing is to ensure that the system fulfils user requirements. Before going live, the newly developed application system should be thoroughly tested. It is the process of executing the application programs with the intent of finding errors. This is achieved using carefully planned test strategies and realistic data so that the entire testing process is methodically and rigorously carried out.

Very few programs perform correctly in the first time they are written. Also, the requirements for the software often change during development. The program must be rewritten thoroughly to make sure that it is working properly and performing.

Testing and debugging are important processes used in system development to discover a defect or bug that is present in the product. Usually these two processes are carried out incrementally in conjunction with system implementation.

Software testing is often reflecting to as verification and validation. Validation makes sure that the developer is building the right product according to the specification and verification checks the quality of the implementation. In this chapter, software testing fundamentals, testing strategies and software debugging methods will be presented.

The objectives of software testing are:

- To reveal different classes of errors and do so with a minimum amount of time and effort.
- To demonstrate that software functions appear to be working according to specification.
- To demonstrate that behavioral and performance requirements appear to have been meet.

7.1 Guidelines used in Debugging

While develops Electronic Medical Records the following guidelines are used for determining the cause of a problem.

- *Find the place where the error occurs*

  Finding the place where the error occurs is crucial for fixing coding problems. If you have a large, complicated program that simply fails, the step is to determine exactly where the failure is occurring. This is not necessarily an easy task, depending on the complexity of the code. The easiest way to find the source of an error is to trace the program as it runs, examining the values of the data structures to determine what is wrong.

- *Define exactly what is wrong*

  Once you know where the problem happens, you need to define exactly what the problem is. Is a calculation return the wrong result? Does the wrong data get inserted into the database? In order to fix a bug, it is necessary to know how the bug manifests itself.

- *Reduce the program to a simple Test Case*

  A good strategy to follow when you don't know where the error is occurring is to reduce the program to a simple test case. Start cutting out parts of the code, and return the program. If the error still occurs, you will know that the section you remove did not cause the error. Examine the section that was remove if the error goes away.

  Remember that one area of our code may have the bug, but the problem may manifest itself in another part of the code. For example, a procedure may return incorrect value but the returned value is not actually used until later in the main program. Cutting out the procedure call and replacing it with a direct assignment to returned value would reveal the source of the problem.

- *Establish a Testing Environment*

  Ideally, testing and debugging is not done in a production environment. It is a good idea to maintain a testing environment that production as much as possible-the same database but with less data. This way you can develop and test newer version's of your application without affecting the production version that is already running.

7.2 Test Data

Test Data was used in the execution of the problem. In Electronic Medical Record, a series of test was conducted with data that are individually designed to represent the real environment as closely as possible. The following categories of test data were used to execute the program.

- *Normal Test Data*

  Testing with normal test data is a procedure whereby the program goes through a light and simple test to determine whether the program runs or not and to determine whether it is error-free.

- *Erroneous Test Data*

  Test with erroneous test data is a procedure whereby the program goes through an erroneous test. Erroneous test is a test where errors are keyed in intentionally. This test is vital to determine how the program or system can handle such errors or incorrect data and from there, the reliability and efficiency of the system can be predicted.

7.3 Documentation Fault

When the documentation does not match what the application does, the application has documentation faults. Usually, documentation is derived from system design and provides a clear description of what the programmer would like to program to do, but the implementation of these functions is faulty. Such faults can lead to other faults later.

7.4 Test Planning

The purpose of having test planning is to help in designing and organizing tests, so that testing is carried out appropriately and thoroughly.

A test plan has the following steps:

1. *Establishing test objectives.*

   At the beginning, we have to know what we are going to test on. So, we have to know establish our test objectives that tell us what kinds of test cases to generate.

2. *Designing test cases*

   After establishing test objectives, we begin to design the test cases that are used to test the system. If test cases are not representative and do not thoroughly exercise the functions that demonstrate the correctness and validity of the system, then the reminder of the testing process is useless.

3. *Writing test cases*

   After designing, we have to start writing the test cases.

4. *Testing test cases*

   At the same time, we also test the test cases.

5. *Executing tests*

   After all testing have been done, we execute our tests on the system

6. *Evaluating test results*

   After executing tests, we evaluate the test result.

7.5 Testing the system

Testing is a process of exercising or evaluating a system by manual or automatic means to verify that it has satisfied requirements or to identify differences expected and actual results. Testing is probably the least understood part of a software development project. A bug is any unexpected, questionable or undesired aspect or behavior displayed, facilitated or caused by the software being tested. Testing can uncover different classes of errors in a minimum amount of time and effort. There are many effective approaches to testing object-oriented software. However, most have been recently developed and are not yet widely disseminated. A test process that compliments object-oriented design and programming can significantly increase reuse, quality, and productivity. The strategies used for testing are unit testing, integration testing and system testing.

7.5.1 Unit Testing

Unit testing verifies that the component functions properly with the types of input expected from studying the component's design. The first step is to examine the program code by reading through it, trying to spot algorithm, data and syntax faults. This is followed by comparing the code with specifications and with the design to make sure that all relevant cases have been considered. Next, the unit module is compiled and executed to view form or result and then eliminate remaining syntax faults if necessary. Finally, test cases are developed to show that the input is properly converted to the desired output.

The first stage of testing Electronic Medical Records is unit testing. Unit testing involves testing each component on its own, isolated from the other components in the

application. The following steps specify how unit testing is carried out for this application:

1. The code of the program is examined by reading through it to spot for algorithmic fault and syntax fault.
2. The unit module is compiled and executed to eliminate remaining faults.
3. Test cases are developed to ensure that the input is properly converted into the desired output.

Choosing test cases

To test a component, input data and condition are chosen. Then, the component is allowed to manipulate the data, and output is observed. The input is selected so that the output demonstrates something about the behavior of the code. A test point or test cases is a particular choice of input data to be used in testing a program.

To perform tests on the components, we must first determine the test objectives. Then, we select test cases and define a test designed to meet the specific objective. Some data are purposely chosen to be improper. This is to check that the code handles incorrect data gracefully.

Test thoroughness

To test a code thoroughly, we can choose test cases using at least one of several approached based on the data manipulated by the code:

1. Statement testing: Every statement in the component is executed at least once in some test.

2. Path testing: Every distinct path through the code is executed at least once in some test.

Unit Test Case Example

Each table in database has at least associated with one-trigger programs. Unit testing was carried out on each trigger program once it was completed. Table below shows the test case for unit testing on the administrator maintenance trigger program.

Table 7.1 Test cases for administrator maintenance

Step	Test Procedure	Expected Output	Test Result Analyzing
1	Add a new patient particular into the patient information form and *submit* button	The record is inserted into database permanently	Record is inserted successfully
2	Press the *view patient database* to view the patient information	Patient's information will display out	The information is display correctly

7.5.2 Integration Testing

When the individual components are working correctly and meet the objectives, these components are combined into a working system. In other words, integration testing is the process of verifying that the system components work together as described in the system and program design specifications.

Integration testing is used in Electronic Medical Record for constructing its program structure while at the same time conducting tests to uncover errors associated with interfacing. The objectives are to take unit-tested modules and build a program structure that has been dictated by design. This testing will ensure that the interfaces such as the module calling sequence in Electronic Medical Record are systematized and link to the correct document.

In Electronic Medical Record, an incremental integration strategy approach is used. Electronic Medical Record main system is constructed and testing in small segments, where errors are easier to isolate and correct; interfaces are more likely to be tested completely.

7.5.3 Sub-system Testing

This phase involves testing collections of modules that have been integrated into sub-systems. Sub-systems may be independently designed and implemented. The most common problem that rose in Electronic Medical Record is sub-system unable to handle

data type mismatch and causing run-time error. The sub-system test process was therefore concentrated on the exception error handling.

Sub-system Test Case Example

After administrator has successfully login into Electronic Medical Record, he or she can now choose what menu that wanted to be generate, either password changing, adding new patient, view patient's record, billing system, or medical certificate system. Table below shows the test case for generating administrator module.

Step	Test Procedure	Expected Output	Test Result Analyzing
1.	At Electronic Medical Record Administrator main menu, select *new patient*	A form of patient's information interface will pop out	A form of patient's information interface appears.
2.	Select *submit* tab	Generate *new patient* page will be displayed	The correct page was successfully shown
3.	Select *view patient* tab	Preview of patient information will display	Patient information with correct data display out.
4.	Select the *patient database* tab	Generate xml database with patient information	The xml database output of patient information was successfully display out.

Table 7.2 Test case for generating administrator module

7.5.4 System Testing

The last testing procedure done is system testing. Testing the system is very different from unit testing and integration testing. The objectives of unit testing and integration testing are to ensure that the code has implemented the design properly. In other words, the code is written to do what the design specifications intended. In system testing, a very different objective is to be achieved, that is to ensure that the system does what the users want it to do.

Electronic Medical Record is tested whether it meets specific performance efficiency objectives in performance testing. Data Integrity Testing is used to verify that the data is stored in a manner where it is not compromised under updating, restoration or retrieved processing in Electronic Medical Record.

The following system testing was carried out:

The last testing procedure done is system testing. Testing the system is very different from unit testing and integration testing. The objectives of unit testing and integration testing are to ensure that the code has implemented the design properly. In other words, the code is written to do what the design specifications intended. In system testing, a very different objective is to be achieved, that is to ensure that the system does what the users want it to do.

Electronic Medical Record is tested whether it meets specific performance efficiency objectives in performance testing. Data Integrity Testing is used to verify that

the data is stored in a manner where it is not compromised under updating restoration or retrieval processing in Electronic Medical Record.

The following system testing was carried out:

1. **Recovery test**

   Recovery test address responses to the presence of faults or loss of data, power, devices or services. One of these was carried out by restore the database wills the application is running. The test result showed that Electronic Medical Record application could still function properly by retrieving records from the database.

2. **Stress Test**

   Stress test is to determine whether a program fulfill the requirements defined for it. Equally important is to make sure that program works, as it should, even under extreme condition.

3. **Security Testing**

   Verify the protection mechanism in the system against improper penetration.

4. **Performance Testing**

   Performance Testing addresses the non-functional requirements of the application. The type of performance tests carried out for this application are:-

   I. Volume Tests

   The fields and records are checked to see if they can accommodate all expected data.

II. *Security Tests*

This test ensures that the application fulfills the security requirements.

III. *Timing Tests*

System performance is timed to ensure that it meets user's requirements.

IV. *Human Factor Tests*

Display of the page and messages are examined to determine user friendliness.

5. Usability Test

This test is on attempt to find human-factor or usability problems.

7.6 Summary

Testing is one of the important steps in developing a system. Precision and accuracy of output data is considered during this process. Unit, module, integration and system testing has been carried out for the Electronic Medical Record. These testing approaches lead to delivering a quality system to users. The objective of a system will only achieve after all the thorough testing done by different user with different aspects.

CHAPTER 8:SYSTEM EVALUATION

8.0 Introduction

System evaluation is a process where system developer evaluates the system after the system has been fully developed. Normally, the developer will evaluate the system from many aspects, which will summarize the system strengths and limitation. In addition, the problems encountered during process developing were listed down with the solutions figure out during the developing process. This chapter also includes the future enhancements for the system in order to get a more satisfactory system.

8.1 Problem Encountered and solution

The following are the major problems encountered from the beginning of the project through the end of the system development process.

- **Difficulties in choosing a Development Technology, Programming Language and Tools.**

  There are many software tools available to develop a window-based database system currently as stated in the earlier chapters. Choosing a suitable technology and tool was a critical process as all tools have their strengths and weaknesses. In addition, the availability of the required tools for development is also a major consideration.

  Solution:

  In order to solve this problem, advices and views were sought from project supervisor, course mates and even seniors engaging in similar project.

Furthermore, surfing the internet and visiting the library helped to clarify some doubts.

- **Set up the Development Environment**

The operating system needed to run this project is Microsoft Windows XP Professional. So, the computer has to be formatted. Furthermore, the set up of the server are critical for the operation of the application development. Therefore, the setup process took a long time because lack of experience. Besides, the repeated failure of the server required re-installation as a remedy and this consumed time and effort.

Solution:

This difficulty is solves with the assistance of my friends, those who had experienced to set up such development environment.

- **Determining scope of system to be built**

As the name XML data warehouse for browser-based electronic medical records implied, to build a full scale complete system is impossible within the given time frame. Inexperienced with the current survey procedure in the real world was another hindrance to implement time workable registration procedures as being implement now in the industry.

Solution:

Many discussions were held with project supervisor to outline the scope of project to be built during in the initial stages of the project. After the scope has been defined, analysis of the others almost same system was done.

- **Lack of evaluation by the end user.**

It was unable to look for a hospital group to test my system. I do not really know how and user or administrator will react to this system.

Solution:

As a result, my course mates and friends help me did the testing process.

- **Insufficient Development Time**

There are still a lot of features that can be build into Electronic Medical Record to make it more useful and towards to achieved the goal of handling XML data warehouse for browser-based. Due to the time constraint, Electronic Medical Record only implemented on crucial features and functionality, before any additional tools are develops.

Solution:

Time given to us to develop a system is very short since that we still have handle assignments that given to us by others subjects. For me, there is not a good solution can suggest because all the student have to do so to graduate.

- **Insufficient knowledge on Electronic Medical rules, hospital work flow and XML application**

  As a beginner, building Electronic Medical Record application with development software is not an easy job. This is a major problem as the hospital's work flows and business rules in the real world is somehow different from the daily seen stand-alone application programming. As a result, an exhaustive effort was done on scratching the basic understanding and skills that are necessary to meet the end-user's requirements.

  Solution:

  Getting hospital information and also XML notes through internet were among some of the efforts taken to overcome this problem.

- **Lack of knowledge Visual Interdev, Macromedia Dream weaver MX and IIS.**

  Since there was little knowledge of programming in Visual Interdev, dream weaver MX and also IIS application, there was an uncertainty in programming on how to update it in the local host. Besides that, all that were never been taught.

  Solution:

  Discussion with friends using the same technology was great help. A more efficient step was through trial and error during the coding phrase.

8.2 System strength

During the development of this project, several system strengths were identified and described as follows:

- **User friendliness and Easy to use Interface**

  Some useful Graphical User Interface (GUI) such as icon, command buttons, drop down list boxes and record sorting button are provided which attract the users to navigate through the system and give faster access. This user-friendly interface can shorten the learning curve and reduce training costs, which include money and time. The menu-driven and pop-up windows or pull-down menus are built to facilitate the individual needs of the users.

- **Security Features**

  Security issues are taken into consideration for the each and every module so as to prevent any unauthorized users from manipulating the data stored in the database. The maintenance module is protected with password and specific login ID. Only the authorized person is allowed to update and add data into the database.

- **Display processing message**

  There are a lot processes between the system and its database, therefore, it is important to inform the users what the system has done when clicking on any command button. For example, message like "information is saved successfully" will be displayed to inform the user. This is to let the user knows that the record is already inserted or deleted into/from the database successfully. Without these messages user might think that the system is faulty if no results are returned when a command is executed.

- **Able to provide database maintenance and utilities**

  Users are able to do housekeeping for database maintenance. They can create, add, modify, update and delete the association information. Besides that, they can also keep track on the records and view the reports graphically. Furthermore, the user can back up and restore the database.

- **Useful Report Generating**

  Electronic Medical Record able to generate useful patient information, billings and medical certificate (MC).

8.3 System Limitations

The following list the limitations of the Electronic Medical Record system:

- *Installation limitation*

  Because of Electronic Medical Record was developed using Visual Interdev and Dream weaver MX, so it can only run properly under Windows 2000, Windows ME and Windows Professional. In order to install Electronic Medical Record in earlier version of Windows operating system, some installation package need to be download from Microsoft website and some setting used to be performed. For further information, please kindly refer to Microsoft website, http://www.microsoft.com

- *Mail system*

  Due to the time constraint, internet mail system is not provided in Electronic Medical Record. So, doctors and patient can't communicate with each other through e-mail. This may cause inconvenient to all doctors and patient.

- *User Group Limitation*

  In Electronic Medical Record, it different from others hospital system. Only user group that allowed by the administrator can use this system, which are doctors and patient. But doctors can allowed their assistant to use the system, so it depends to the doctors.

- *Medical Record*

  Due to time constraint, the medical records were not properly prepared into medical format. This is because of the medical records have to be presented into XML data ware house so that times needed in order to explore how to make it work.

- *Password*

  Password is not encrypted and this result in high vulnerability as password.

8.4 Future Enchantments

Future enchantments can be done to make the system more advance and easy to use. A system development knows no boundaries as new requirements and better implementation methods continue to arise and evolve. There are several enchantments that could extend after developed the system:

- *Mail system and Internet Dialer*

  As stated, Electronic Medical Record is not provides send e-mail and check e-mail functions. The system will become more complete if it has mail system and also internet dialer. Internet dialer is use for building Internet connection.

- *Attractive Interface*

  Electronic Medical Record will become better if its interface is enhanced to be more attractive and interactive by adding more meaningful and user-friendly images, 3D images, animation images and sounds.

- *Tool Tip Help*

  Tool tip help will be shown to the user to provide a basic guide to use the part of the form or interface. As an example, tool tip for the delete icon is "click to delete the patient's information record". This makes the user clearly know the action will be taken by the system once they click to the delete image link. So, Electronic Medical Record will be become more user-friendly if have tool tip help.

8.5 Summary

Overall, the Electronic Medical Record system has achieved and fulfilled the objectives and requirements as a hospital application as determined during system analysis. Any hospital or clinic group can use this system to manage their medical records into different form from others. This system will also help hospital and clinic group to keep their medical records in more secured and different ways.

There was a lot of knowledge gained throughout the development of this system. This includes knowledge in hospital workflow, medical rules, XML database, programming and so on. XML database proved to be a valuable experience. Even though programming skills and techniques are important in development, good software engineering techniques must also be applied. Here, theories and knowledge gained throughout the course of Information Technology studies like system analysis and design methods, database, software engineering and software requirements were literally put into practice.

Finally, there are much more rooms for improvement in this system, especially in term of implementing a more satisfactory system. With the first step taken, enhancements could still be made with more features added for future version.

REFERENCE :

Sommerville, Ian (2001). *Software Engineering*. (Edisi ke-6). London: Addison Wesly

Whitten, Jeffrey L.Bently, Lonnie D. Dittman, & Kevin C. (2002). *System Analysis And Design Methods*. (Edisi ke-5). New York: McGraw-Hill.

Mohamad Noorman Masrek, SafariAbdul Rahman, & Kamarulkariffin Abdul Jalil (2001).*Analisis & Rekabentuk Sistem Maklumat*. Kuala Lumpur: McGraw-Hill.

XML Documents and Frequently Asked Questions website

(URL- http://www.ucc.ie/xml/fac.html)

Introduction to XML tutorial website

(URL- http://www.xml.com)

Barbara C. McNurlin. (2002), *Information System Management-5th Edition* New Jersey: Prentice Hall

George M. Marakas (1999), *Decision Support System-In the 21st century* New Jersey: Prentice Hall

Schweiger R, Buerkle T, Ruan W, Dudeck J (1998): *XML: Evolution towards a structured electronic patient record. Electronic Patient Records in Medical Practice, Proceedings of IMIA Working Group 17, Rotterdam 8-10 October 1998.*

Microsoft.com - Web and Application Services(2000). [Online].

Available HTTP:

World Wide Web Consortium: XSL Transformations (XSLT) Version 1.0. W3C Recommendation 16 November 1999.

http://www.microsoft.com/windows2000/technologies/web/default.asp

http://www.xml.com

http://www.microsoft.com

http://www.swynk.com/friends/Vartanyan/Access_vs_SQLServer.asp

http://searchwin2000.techtarget.com

ADMIN MANUAL

1.0 Introduction

Electronic Medical records is a system that is used to simplify day to day task in running a clinic group in a modern hospital at the same time provide management with the necessary information to make strategic decisions. It serves essentially as a medium for communication between the diverse collaborating functional subsystems in a group of clinic in a hospital.

Electronic medical record summaries are now being deployed, using XML to represent, store and display information that is extracted from existing hospital systems. Patient records are created and maintained as XML documents within an XML database. Data feeds from the Hospital's Information System (HIS) make use of the existing HIS interfaces with some customization to meet the data requirements of the XML patient record database. HIS extracts are triggered by events occurring within the system. As updates take place, the XML database can provide full auditing of transactions that have taken place. Access to the browser application is controlled by a security layer that can be integrated with the HIS system. The Web server provides the middle layer between the user interacting with the browser and the XML database.

XML for browser based Electronic Medical Record actually consist of 2 main modules which are:

i) Admin module

ii) User module:

(a) Patients

(b) Doctors

As a Computer Science background student standing point of view, I view the system as a new approach of computer automated hospital information system which supports the various hospital needs of an hospital organization In technical terms wise, the development of XML for browser-based Electronic Medical Record is deemed as a new dimension or more to say a bold step in developing a hospital information system which able to support different organization hospital needs.

2.0 System Specification

The system specification of the XML for browser-based Electronic Medical Record consists mainly of the hardware being used for the client side and the server side.

Server Side Hardware Specification

Hardware requirement refer to the hardware support needed in order to run the system smoothly.

- Pentium III 400MHz computer and above
- 20 GB Hard Disk Space

- At least 128 MB RAM
- Other standard computer peripherals
- Standard Input/Output Devices

Server Side Software Specification

Software Specification refers to the software tools needed in order to run the system in the server

- Extensible Markup Language (XML) 1.0
- Microsoft Access 2003 – Database repository
- Windows NT Server 4.0- Network Operating System
- Microsoft Windows XP Professional
- Microsoft Internet Explorer 5.0
- Macromedia Dream weaver MX
- Notepad

Client Side Hardware Specification

Hardware requirement refer to the hardware support needed for the client side to run the system smoothly.

- A Pentium 3 with 750 MHz processor
- At least a 64 MB RAM
- Network connection through existing network configuration or modem
- Other standard computer peripherals

Client Side Software Specification

Software Specification refers to the software tools needed in order to run the system in the server.

- Microsoft Windows XP
- Microsoft Internet Explorer 5.0

The suitability of the tools being selected is solely based on the compatibility issues being taken into consideration during the design phase. Hence, the selection of the tools is deemed the most suitable to run the system as a web based system.

3.0 XML FOR BROWSER-BASED ELECTRONIC MEDICAL RECORD USER GUIDE

For this part of the User Manual is the most important part focus on helping the user to use XML For Browser-Based Electronic Medical Record which is in fact a very easy to use system which focuses on simplicity in its usage. Basically, this system is categorized into 2 main modules which are admin module and user (patients and doctors) module. So, I will give the user guide of admin module.

3.1 Guide In Using XML For Browser-Based Electronic Medical Record System

Module : Admin

Admin module can be divided into:

I) Change Password

II) New Patient

III) View Patient Database

IV) Modify or Delete a Patient

V) New Bill

VI) View or Delete a Bill

VII) New Certificate

VIII) View or Delete a Certificate

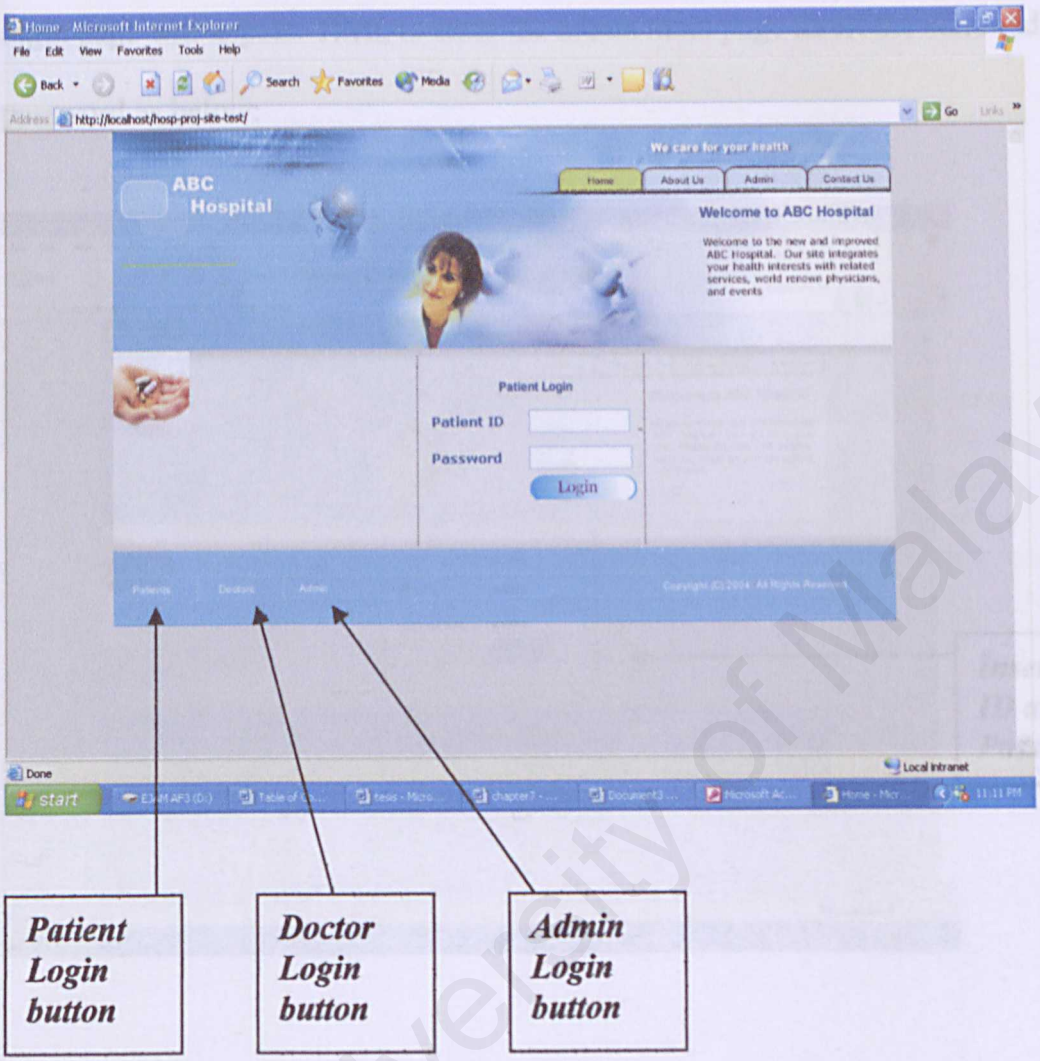
IX) New Doctor

X) View, Modify or Delete a Doctor

With respect to each category, each sub module has their own functions. The following descriptions will focus on the tasks that each sub-modules perform and guides the user on how to use each sub-modules to perform their accounting needs

Main page

Below is a main page once if the admin enter into this system.



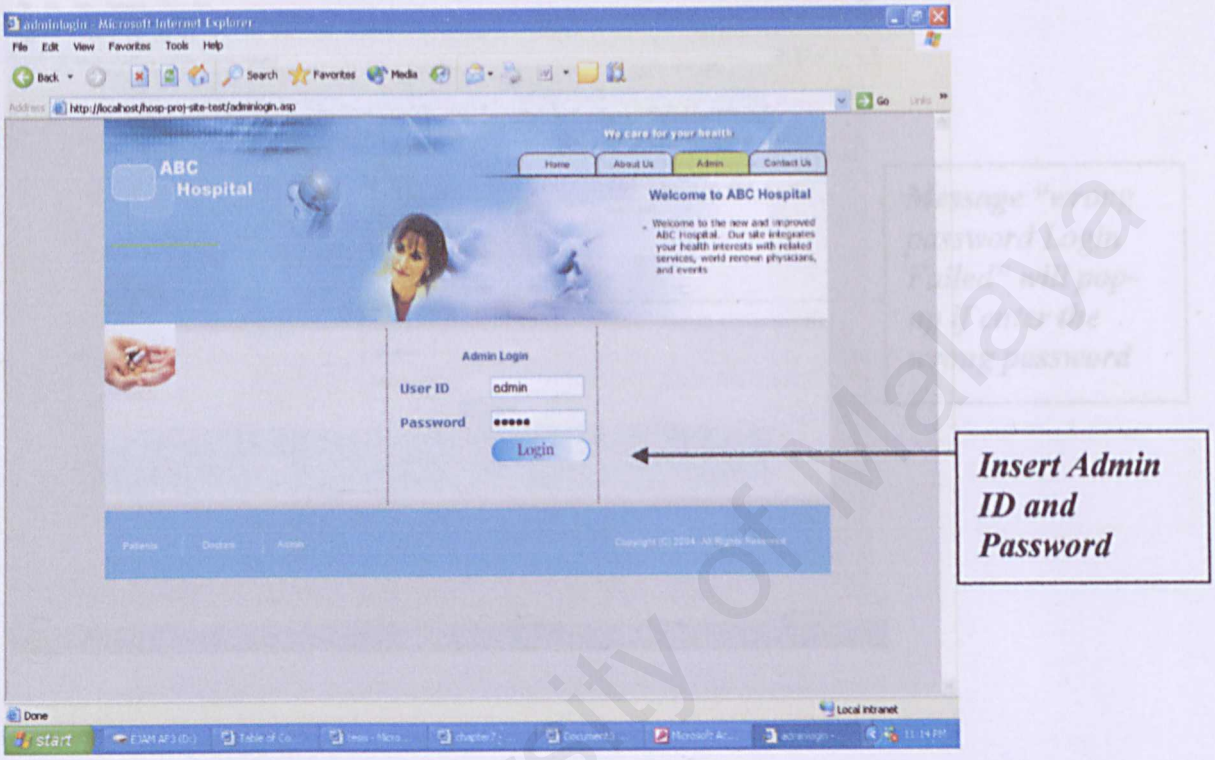
Patient Login button

Doctor Login button

Admin Login button

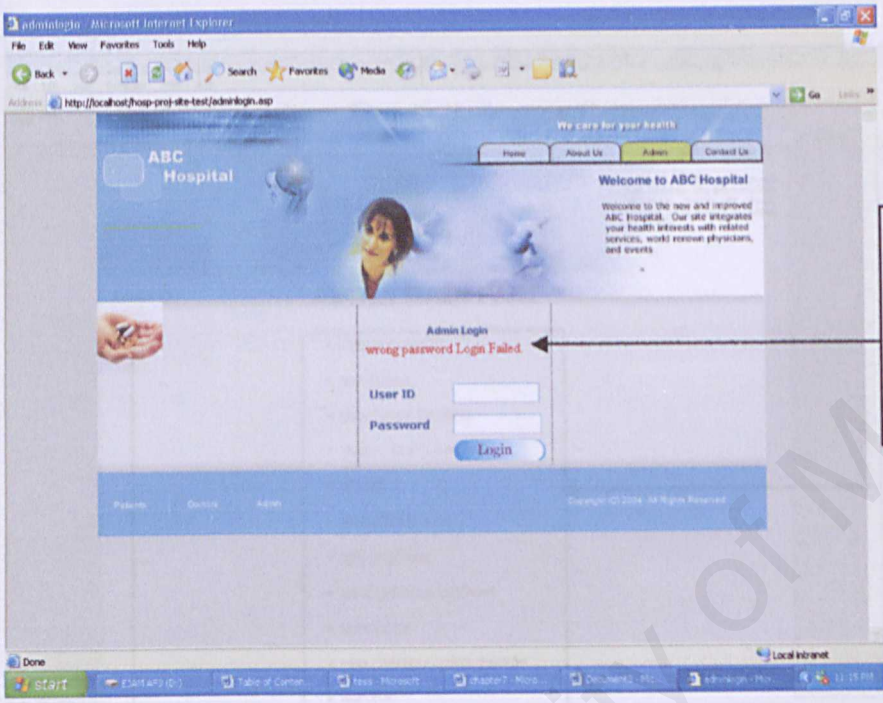
Admin main page

To enter the admin Login page, just click at the admin button and after that the admin login page will appear. Then, to enter the admin main page insert the admin id and password as below:



Admin login : wrong password

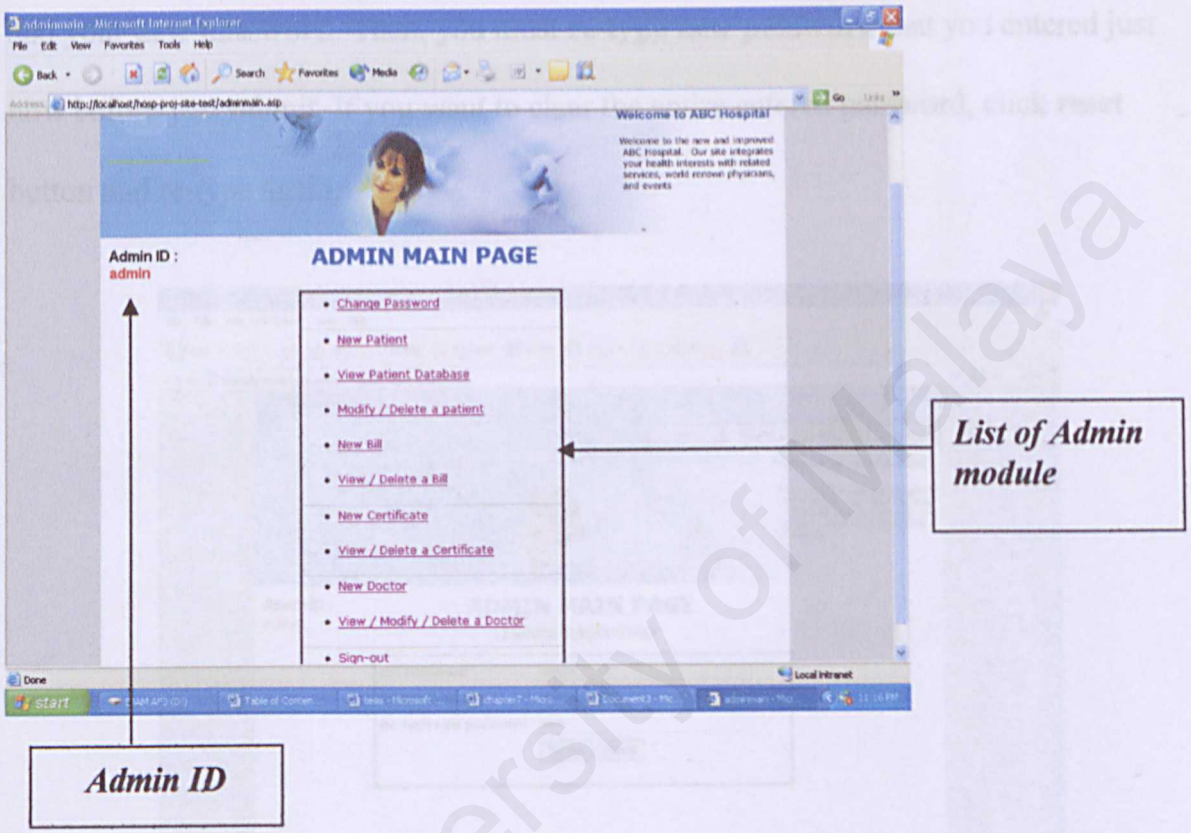
This system is also have its security whereas it also will remind the admin if the password inserted was incorrect. The warning message will appear as below:



Message** "wrong password Login Failed" **will pop-up if enter the wrong password

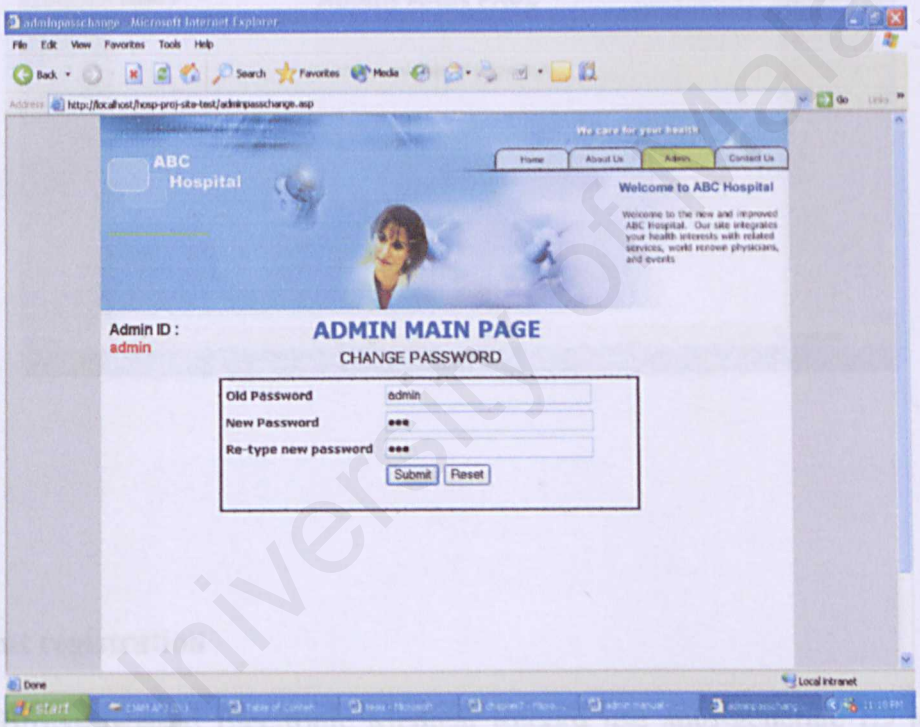
Admin main page

When you have enter into the admin main page, the listing of admin module will appear as below. At the left side of the page, also stated the admin ID.



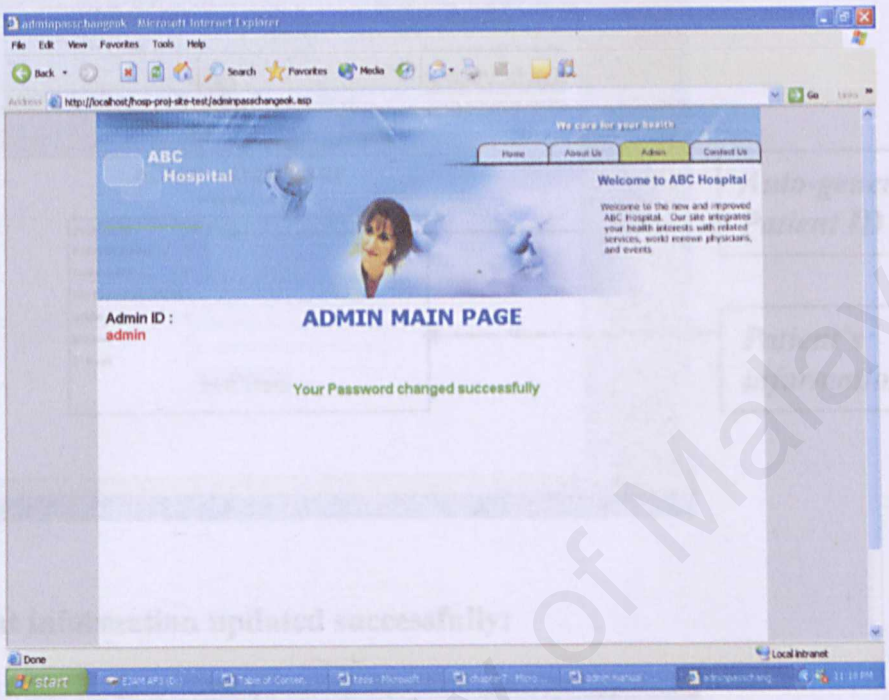
Admin : Change password

If admin want to change password, just click at the **change password** button at the main admin page. After that, a form such as below page will appear. Type the **old password** and your **new password**. Then, you must **re-type new password** that you entered just now before you submit. If you want to clear the entire entered password, click **reset** button and re-type again.



Change password success

If your password you entered is allowed, the message "**Your Password changed successfully**" will pop-up as below.

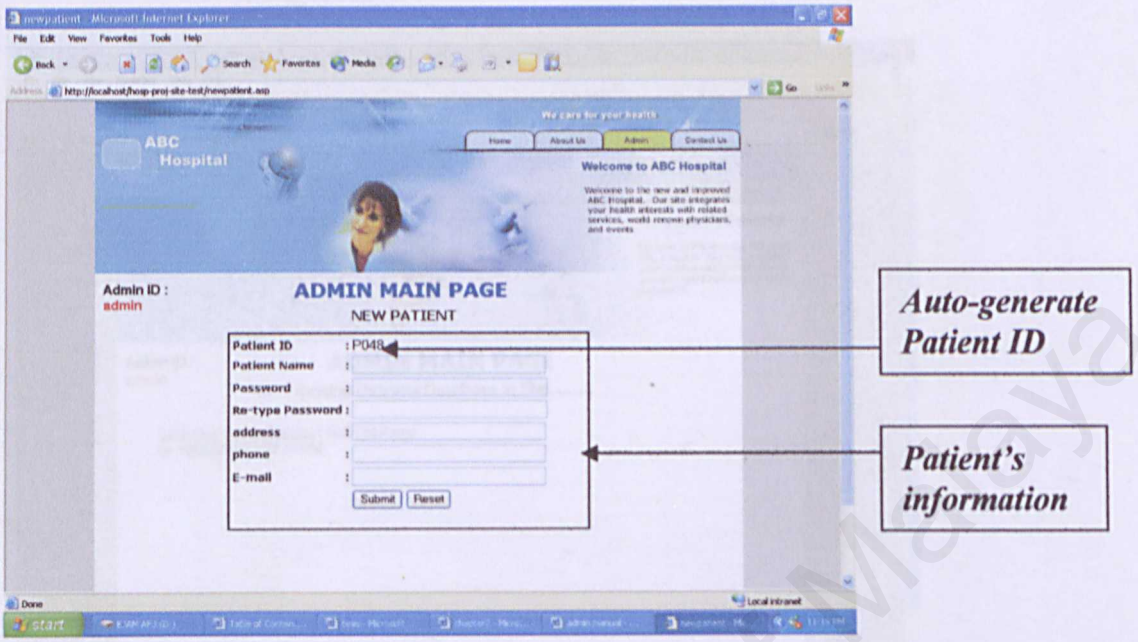


New patient registration

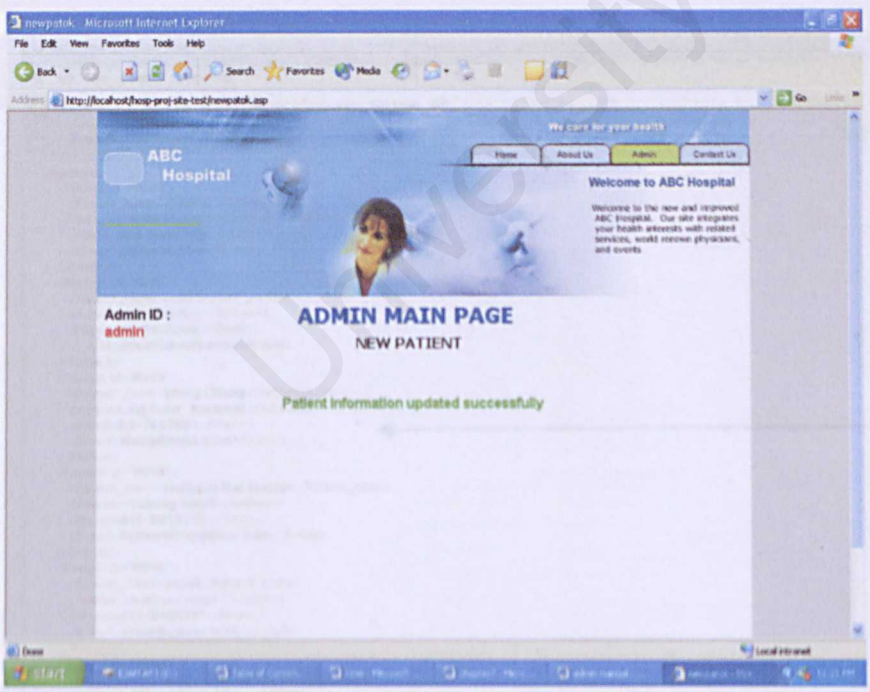
XML for Browser-based Electronic Medical Record use auto-generate ID registration. So, automatically if a new patient wants to register his/her information the auto-generate ID will appear. For example as shown at below page, the patient ID is **P048**. Meaning that, P stands for patient, 04 is 2004 registration and 8 is the number of patient of the year.

So, admin just fill in the form which is actually the patient's information that consist of patient name, password, re-type new password, address, phone number and e-mail. If you

want clear all details that have been inserted, click at "**reset**" button and re-type the information again. If the patient satisfies with the information, click the "**submit**" button.

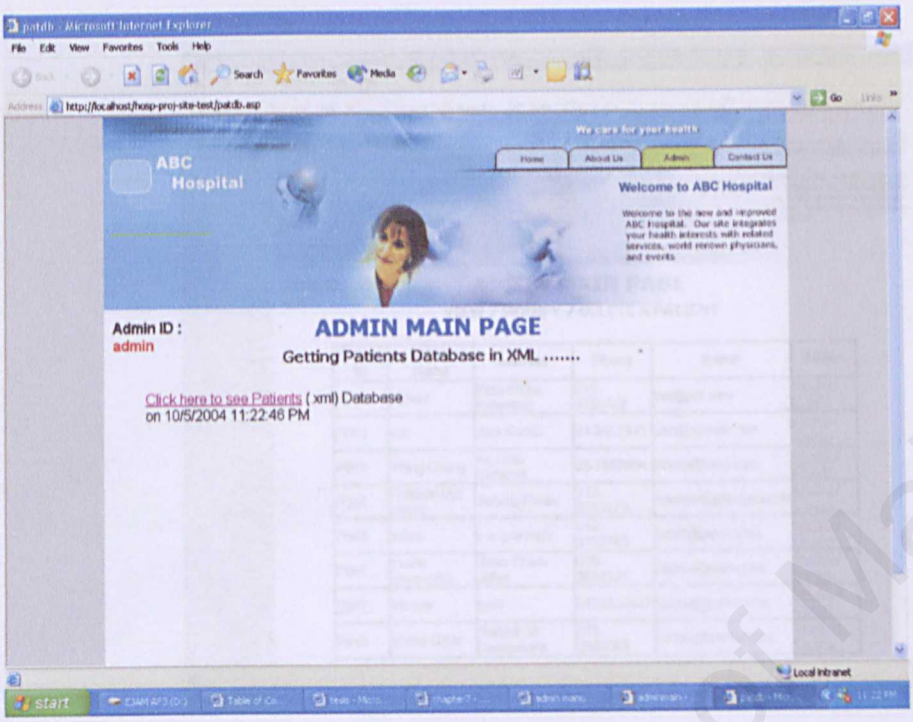


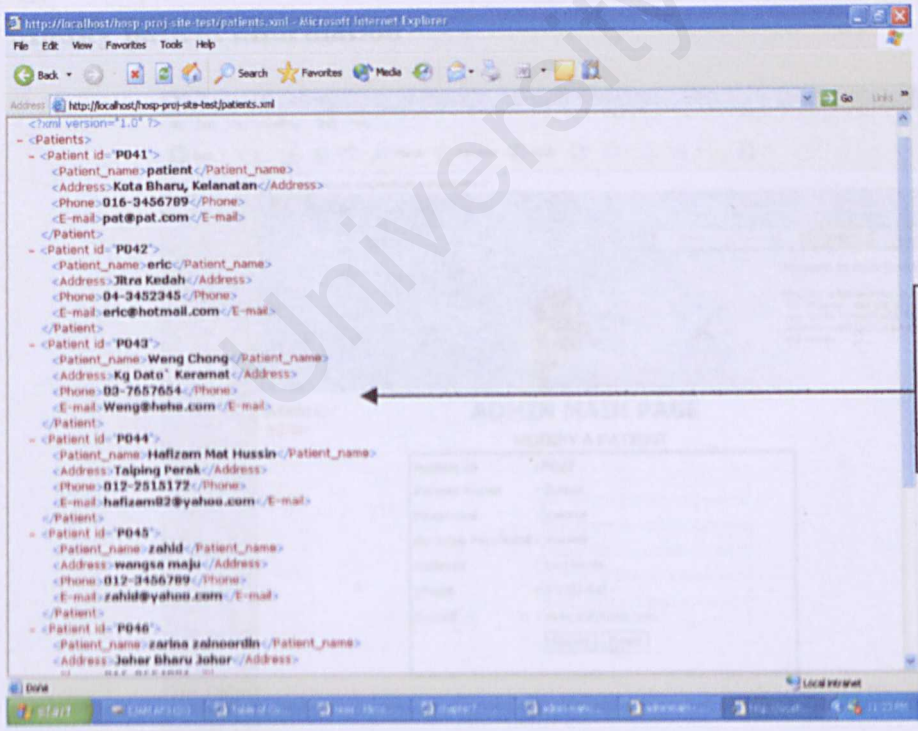
New Patient information updated successfully:



Patient Database in XML format

The special of this system is the features of the patient information database which can be view into XML format.

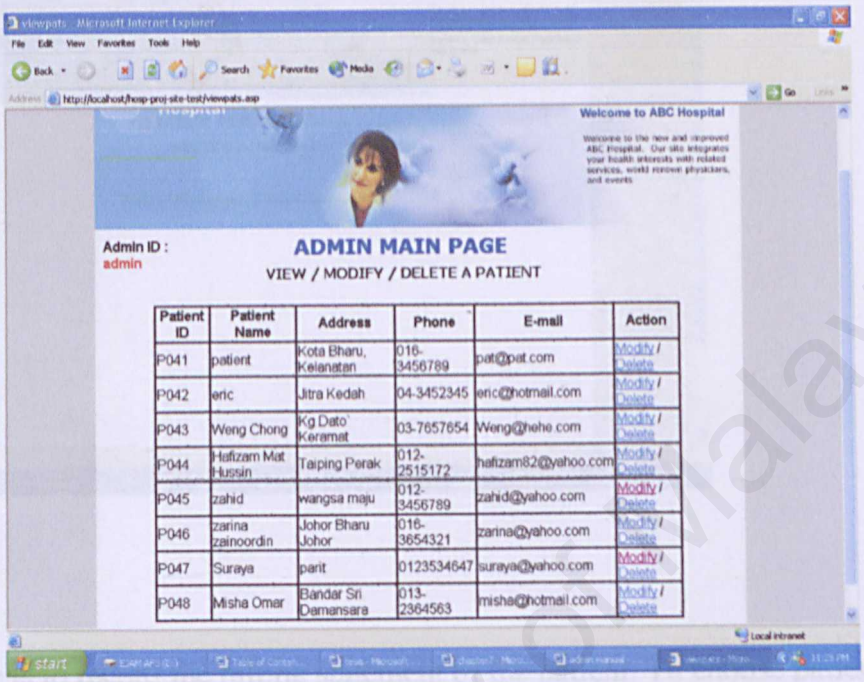




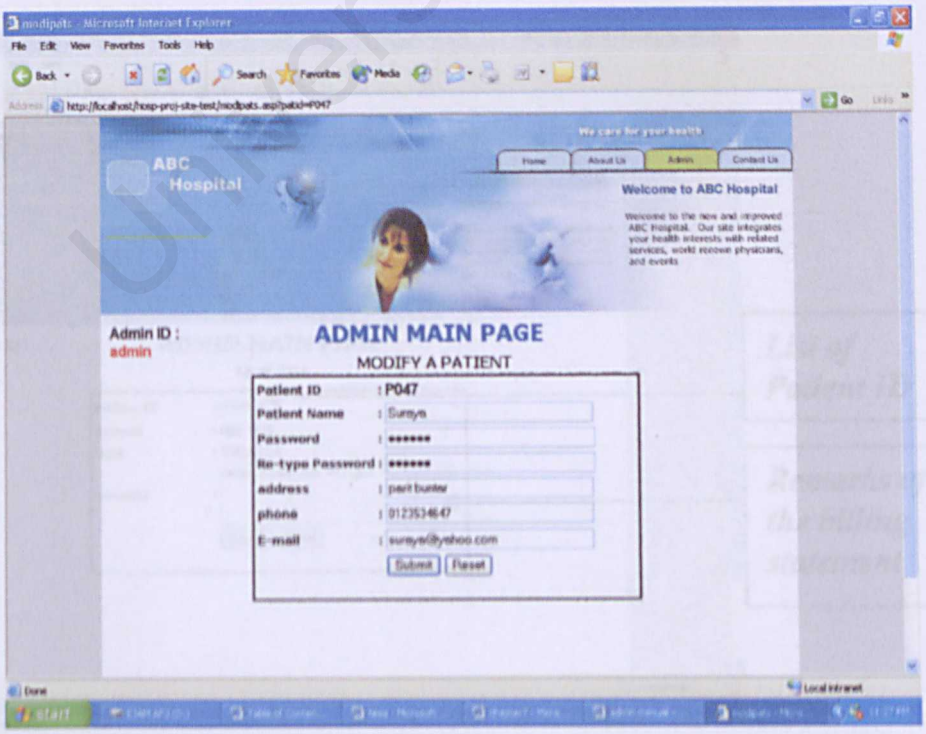
Patient's information database view in XML format

View/Modify/Delete a Patient

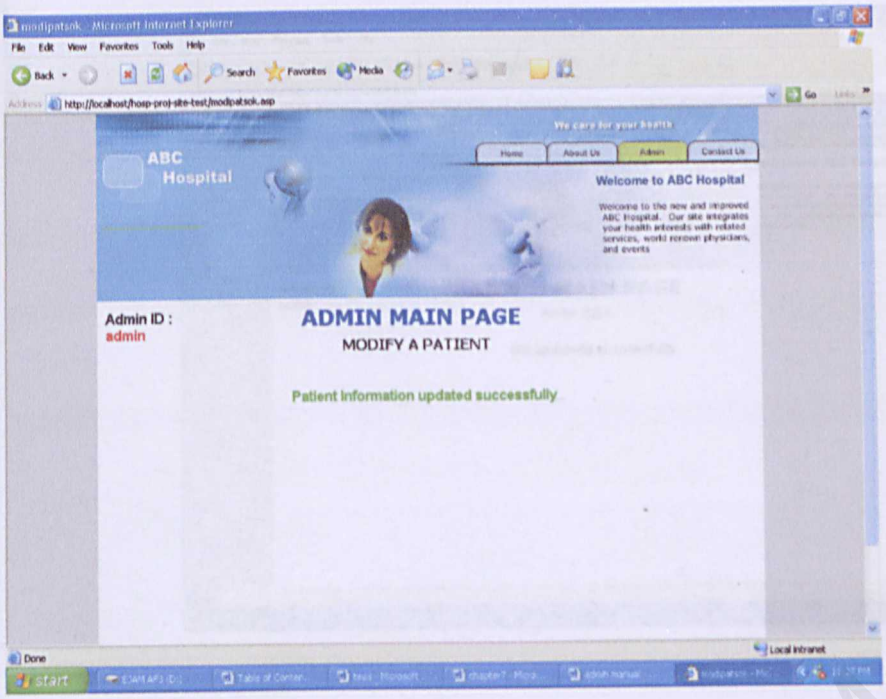
Some of the patient maybe wants to modify such of their information. So, at below page, any information can be modify or even delete and automatically updated.



Modify patient information

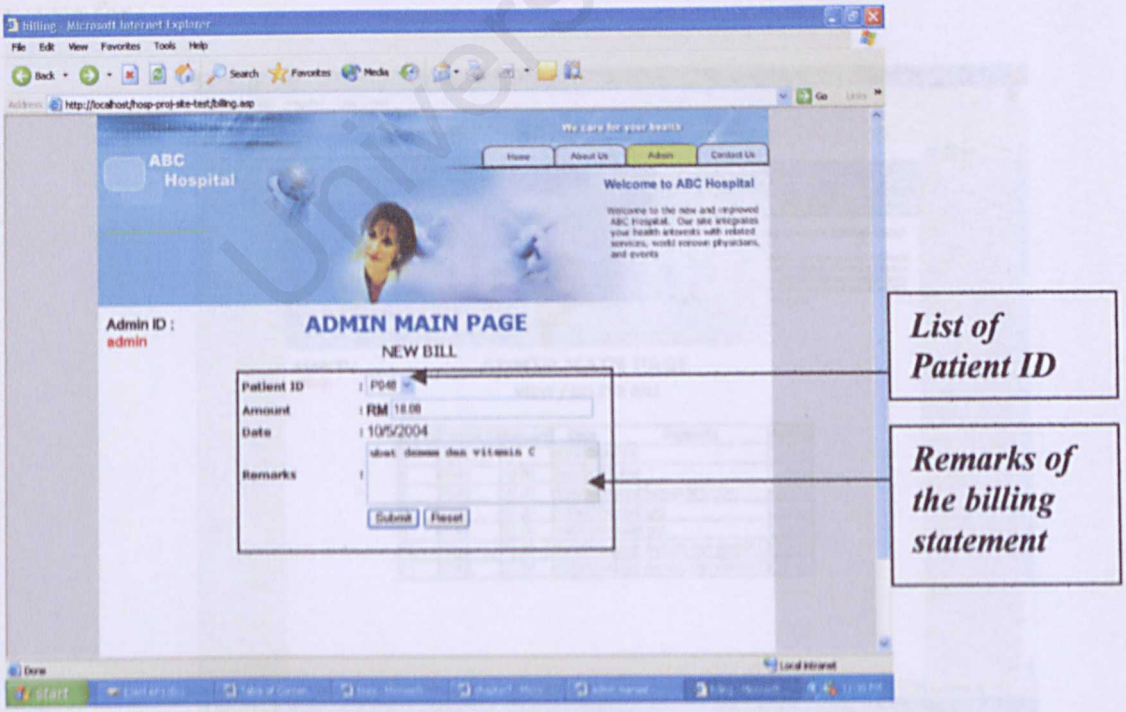


Patient Information Updated Successfully

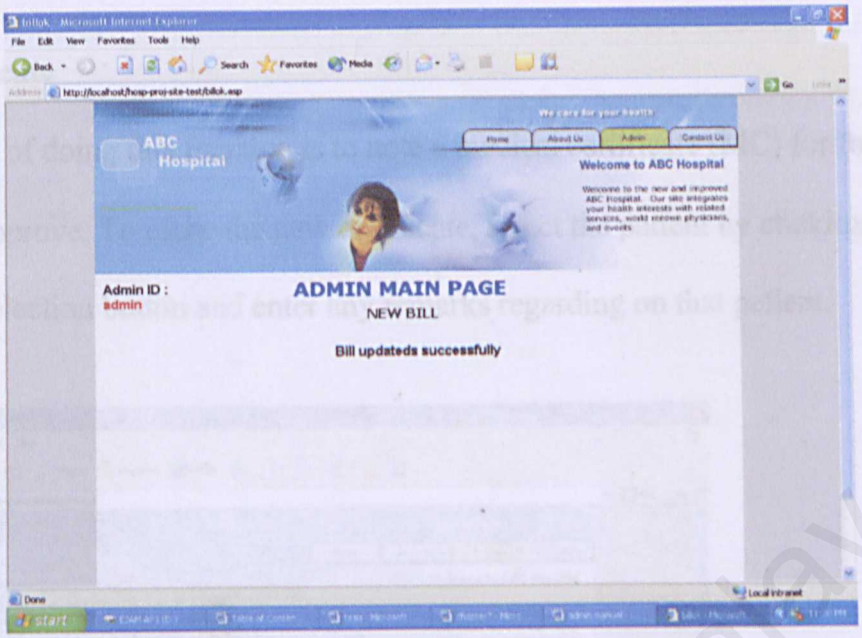


New Bill

This module is to record the billing statement of the patient. To choose patient for new bill, just select from the Patient ID selection and fill in the form before submit.



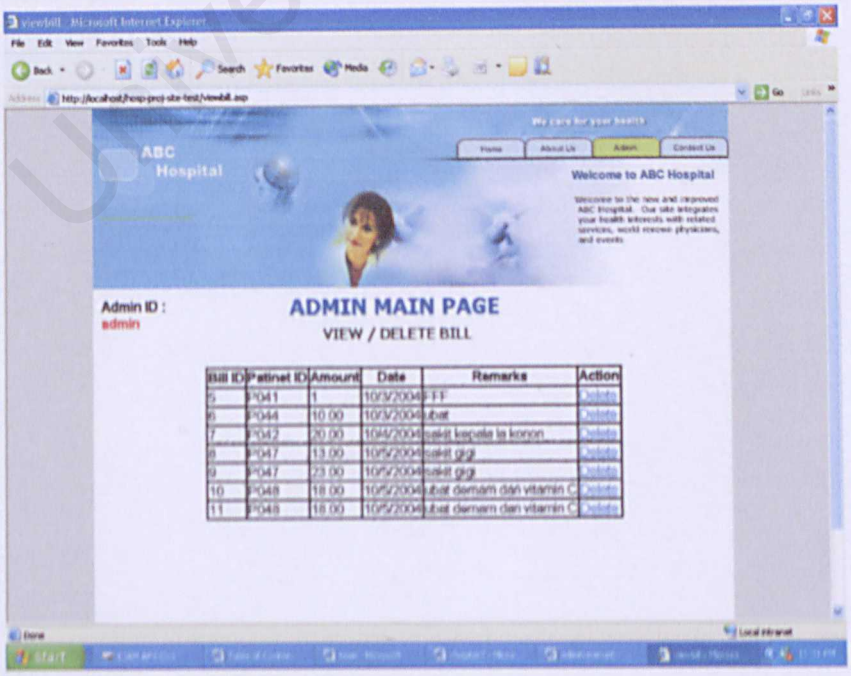
Bill Updated successfully



View/ Delete bill

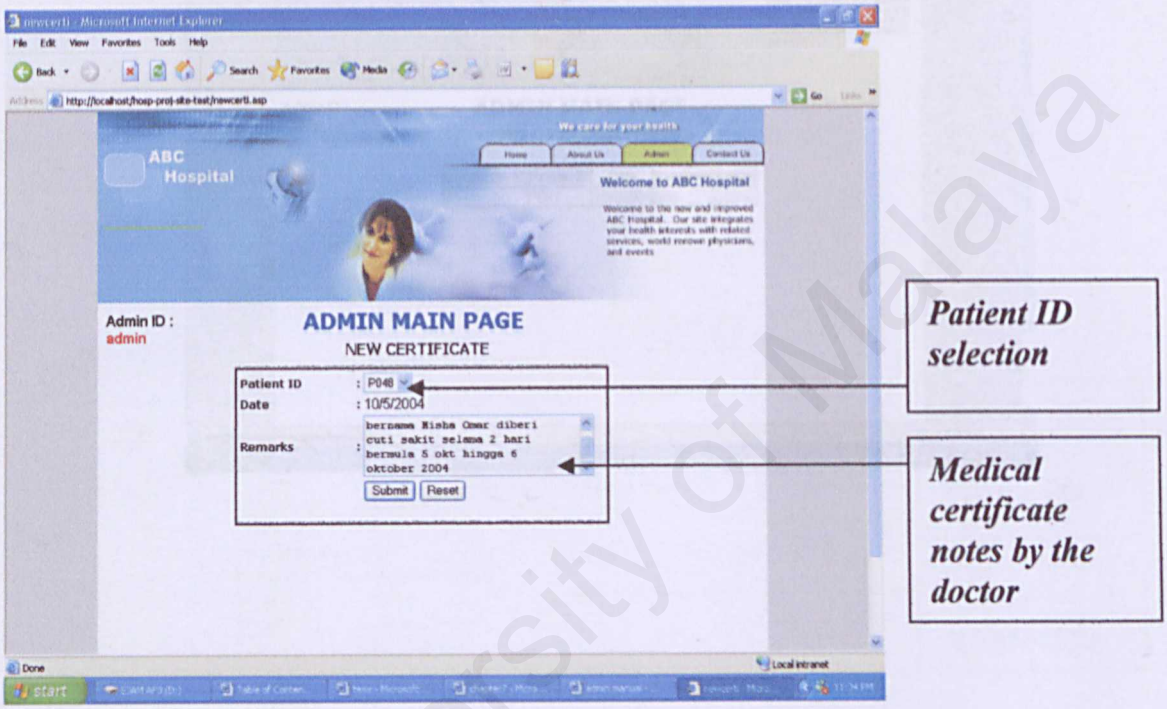
If the admin want to view list of billing statement, click at the ***view/delete bill*** button.

Then, a list of patient's bill will appear and the admin can delete any record that he/she wants for.



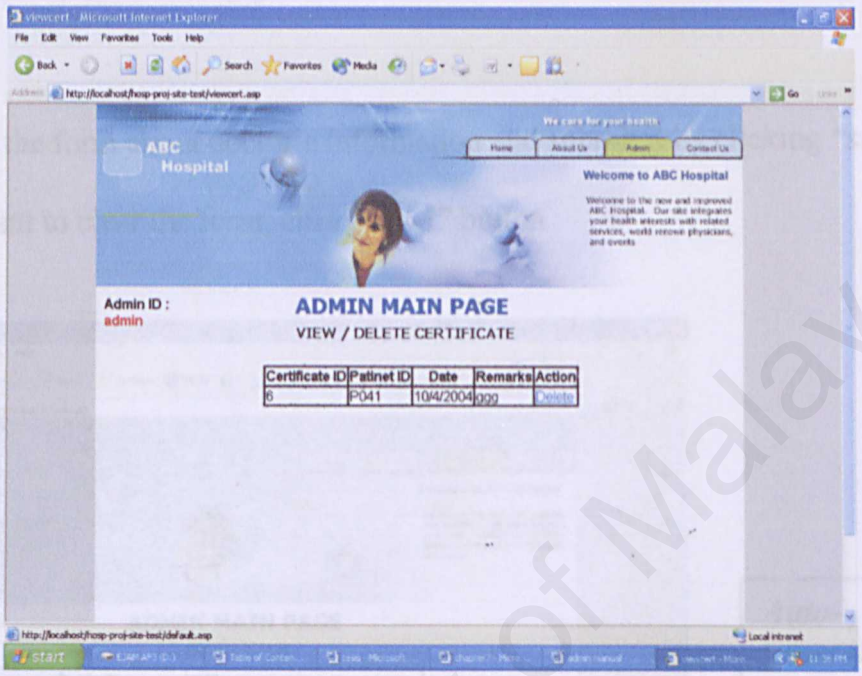
New Certificate

The purpose of doing this function is to note a medical certificate (MC) for the patient if the doctor approve. To make the new certificate, select the patient by clicking at the patient ID selection button and enter any remarks regarding on that patient.



View certificate

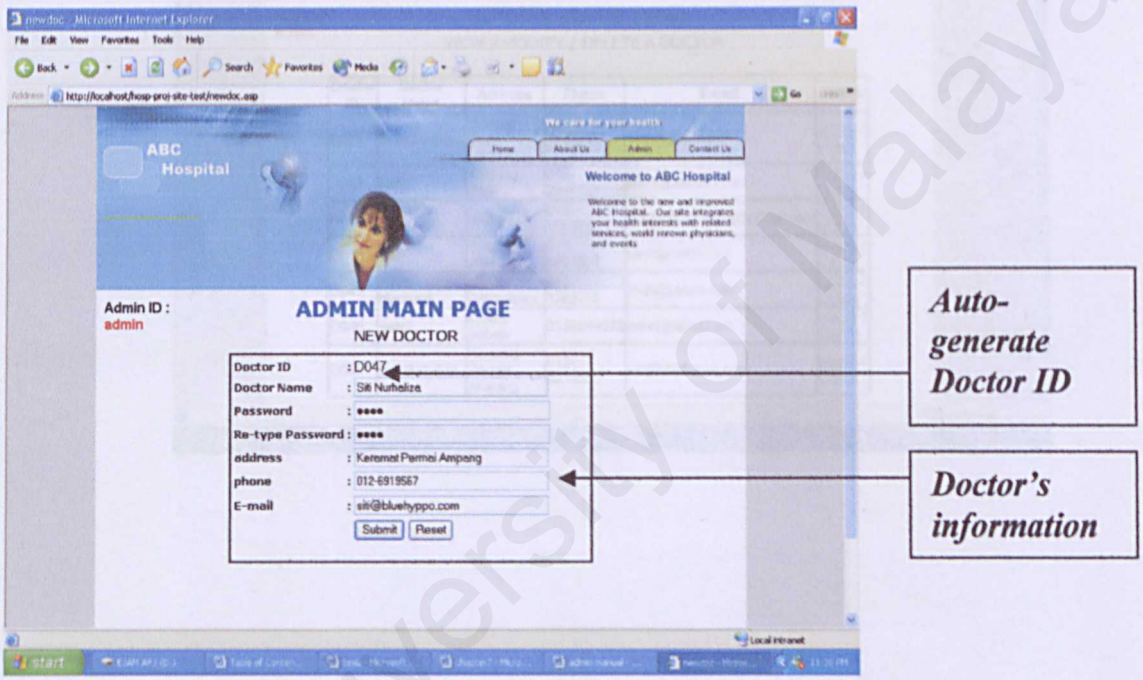
This page is consist of the certificate information that have been made by the admin. If the admin want to clear the certificate information, just click at the ***"delete"*** button.



New Doctor

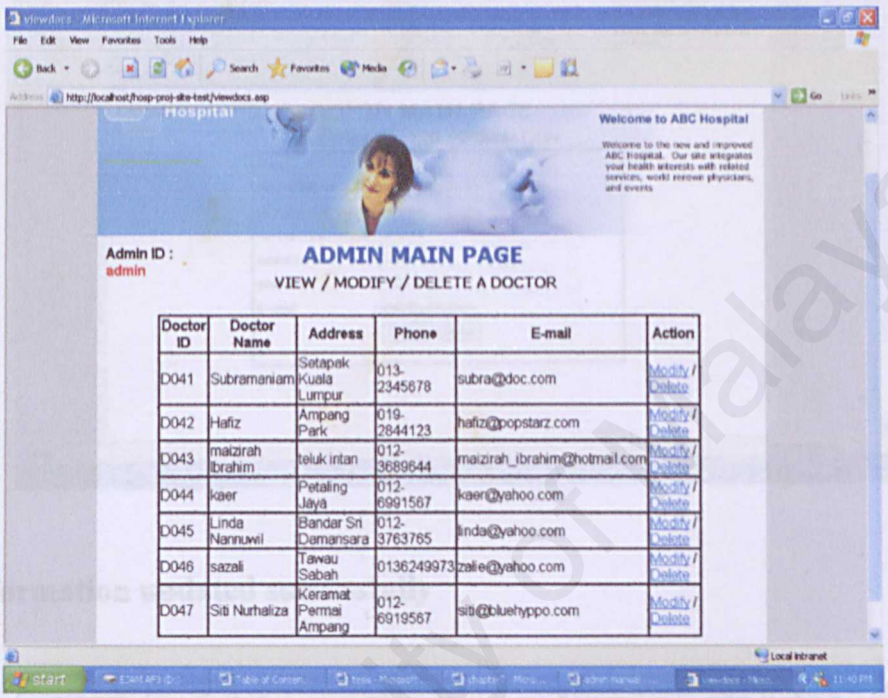
The New Doctor features is basically same as New Patient page. It is also auto-generate ID for new doctor registration. As stated below, the doctor ID is **D047**, meaning that D stands for doctor, 04 is 2004 registration while 7 is the number of doctor register of that year.

Then, fill in the form about doctor's information and submit it by clicking ***"submit"*** button. If want to clear the form, click ***"reset"*** button.

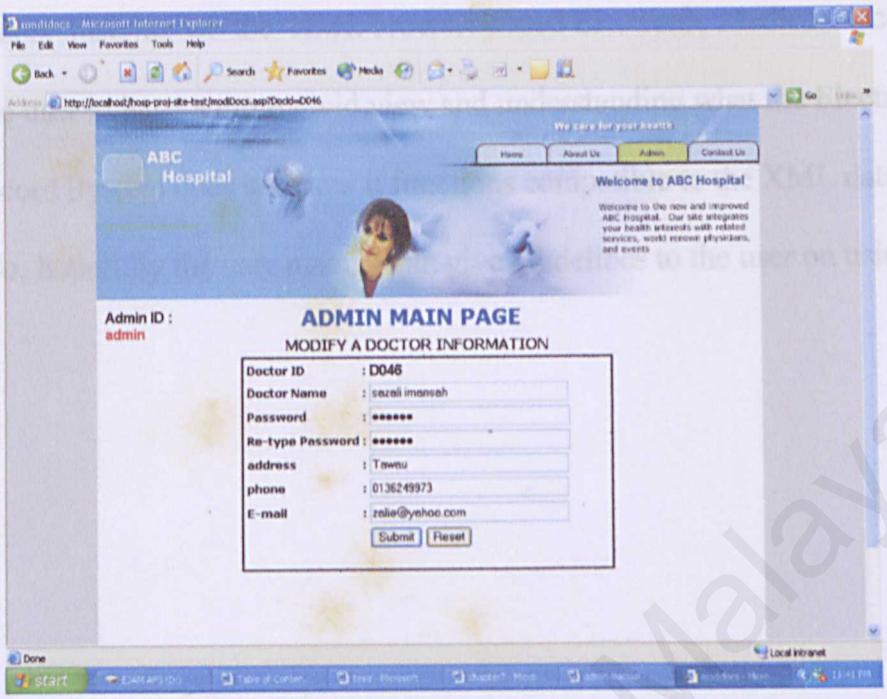


View/ modify/ Delete a doctor

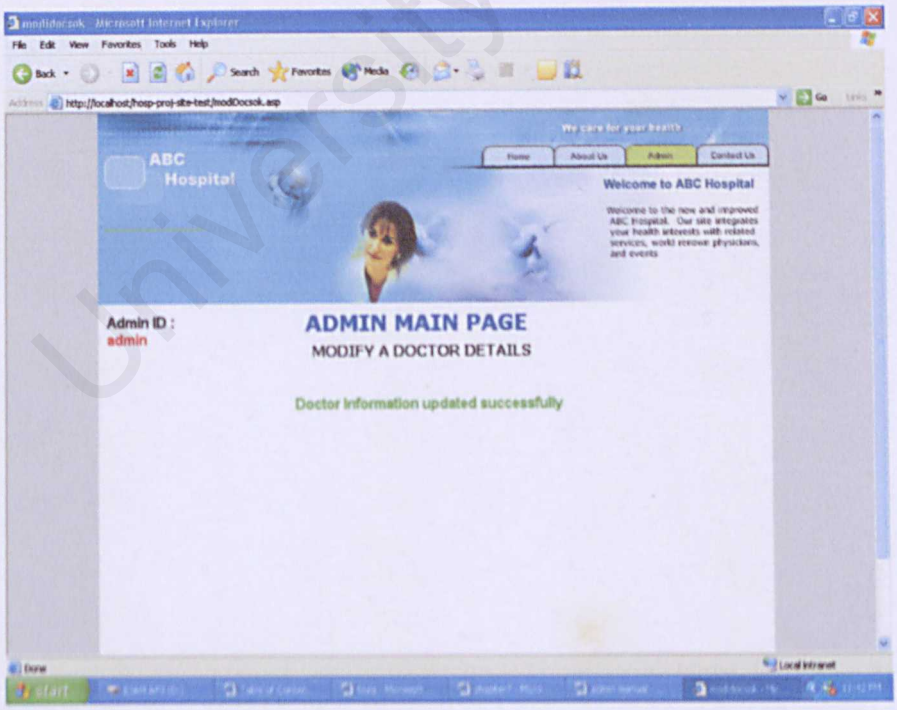
Some of the doctor want to update or modify their information, so by clicking at "modify" button, changes can be made for the doctor's information. Else, admin can also delete the record.



Modify a doctor Information



Doctor information updated successfully



4.0 Summary

The user manual focus like others is to provide the user the most ample knowledge which gives the most insights of the XML browser-based Electronic Medical Record. This is to enable the user to have a more vivid view and understanding what the Electronic Medical Record System does and how it functions compatible to the XML database approach. So, hopefully the user manual can give guidelines to the user on using the system.